

Bonnie Berger (Ed.)

LNBI 6044

# Research in Computational Molecular Biology

14th Annual International Conference, RECOMB 2010  
Lisbon, Portugal, April 2010  
Proceedings

 Springer

# Lecture Notes in Bioinformatics

6044

Edited by S. Istrail, P. Pevzner, and M. Waterman

Editorial Board: A. Apostolico S. Brunak M. Gelfand  
T. Lengauer S. Miyano G. Myers M.-F. Sagot D. Sankoff  
R. Shamir T. Speed M. Vingron W. Wong

Subseries of Lecture Notes in Computer Science

Bonnie Berger (Ed.)

# Research in Computational Molecular Biology

14th Annual International Conference, RECOMB 2010  
Lisbon, Portugal, April 25-28, 2010  
Proceedings



Springer

Series Editors

Sorin Istrail, Brown University, Providence, RI, USA

Pavel Pevzner, University of California, San Diego, CA, USA

Michael Waterman, University of Southern California, Los Angeles, CA, USA

Volume Editor

Bonnie Berger

Massachusetts Institute of Technology

Mathematics Department

Computer Science and Artificial Intelligence Laboratory

77 Massachusetts Avenue, Cambridge, MA 02139, USA

E-mail: bab@mit.edu

Library of Congress Control Number: 2010924885

CR Subject Classification (1998): J.3, H.2.8, I.2, F.2.2, I.6, F.2

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN 0302-9743

ISBN-10 3-642-12682-0 Springer Berlin Heidelberg New York

ISBN-13 978-3-642-12682-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2010

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper 06/3180



# Preface

This volume contains the papers presented at RECOMB 2010: the 14th Annual International Conference on Research in Computational Molecular Biology held in Lisbon, Portugal, during April 25–28, 2010. The RECOMB conference series was started in 1997 by Sorin Istrail, Pavel Pevzner, and Michael Waterman. RECOMB 2010 was hosted by INESC-ID and Instituto Superior Tecnico, organized by a committee chaired by Arlindo Oliveira and took place at the International Fair of Lisbon Meeting Centre.

This year, 36 papers were accepted for presentation out of 176 submissions. The papers presented were selected by the Program Committee (PC) assisted by a number of external reviewers. Each paper was reviewed by three members of the PC, or by external reviewers, and there was an extensive Web-based discussion over a period of two weeks, leading to the final decisions. RECOMB 2010 also introduced a Highlights Track, in which six additional presentations by senior authors were chosen from papers published in 2009. The RECOMB conference series is closely associated with the *Journal of Computational Biology*, which traditionally publishes special issues devoted to presenting full versions of selected conference papers.

RECOMB 2010 invited several distinguished speakers as keynotes and for special sessions on Genomics in Medicine and Regulatory RNAs. Invited speakers included Cecilia Arraiano (Instituto de Tecnologia Química e Biológica Universidade Nova de Lisboa), Chris Bakal (Institute of Cancer Research), David Bartel (MIT, Whitehead Institute and Howard Hughes Medical Institute), Jan Gorodkin (University of Copenhagen), Simon Kasif (Boston University), Isaac Kohane (Harvard Medical School), Doron Lancet (Weizmann Institute), Klaus Lindpainter (Roche Genetics), Patrice Milos (Helicos BioSciences Corporation), Norbert Perrimon (Harvard Medical School and Howard Hughes Medical Institute), Eitan Rubin (Ben Gurion University), and Mona Singh (Princeton University).

RECOMB 2010 was only possible through the dedication and hard work of many individuals and organizations. Special thanks go to the PC and external reviewers for helping to form a high-quality conference program, and the Organizing Committee, coordinated by Ana Teresa Freitas, for hosting the conference and providing the administrative, logistic, and financial support. We also thank our sponsors, including ISCB, FCT and FLAD. Without them the conference would not have been financially viable. We thank the RECOMB Steering Committee, chaired by Martin Vingron, for accepting the challenge of organizing this meeting in Lisbon. Finally, we thank all the authors who contributed papers and posters, as well as the attendees of the conference for their enthusiastic participation.

# Conference Organization

## Program Committee

Tatsuya Akutsu	Kyoto University, Japan
Nancy Amato	Texas A&M University, USA
Joel Bader	Johns Hopkins University, USA
Serafim Batzoglou	Stanford University, USA
Bonnie Berger (Chair)	Massachusetts Institute of Technology, USA
Phil Bradley	Fred Hutchinson Cancer Research Center, USA
Jadwiga Bienkowska	Biogenidec, USA
Michael Brent	Washington University, USA
Michael Brudno	University of Toronto, Canada
Lenore Cowen	Tufts University, USA
Colin Dewey	University of Wisconsin, Madison, USA
Dannie Durand	Carnegie Melon University, USA
Eleazar Eskin	University of California, Los Angeles, USA
Ana Teresa Freitas	INESC-ID / IST, Portugal
James Galagan	Boston University, USA
David Gifford	Massachusetts Institute of Technology, USA
Eran Halperin	University of California, Berkeley, USA
Des Higgins	University College Dublin, Ireland
Trey Ideker	University of California, San Diego, USA
Sorin Istrail	Brown University, USA
Tao Jiang	University of California, Riverside, USA
Simon Kasif	Boston University, USA
Isaac Kohane	Harvard Medical School, USA
Jens Lagergren	Royal Institute of Technology, Sweden
Thomas Lengauer	Max Planck Institute for Informatics, Germany
Michal Linial	Hebrew University, Israel
Satoru Miyano	Tokyo University, Japan
Bernard Moret	Swiss Federal Institutes of Technology, Switzerland
William Noble	University of Washington, USA
Arlindo Oliveira	INESC-ID / IST, Portugal
Dana Pe'er	Columbia University, USA
Pavel Pevzner	University of California, San Diego
Ron Pinter	Technion, Israel

Teresa Przytcka	NIH NCBI, USA
Knut Reinert	Freie Universität Berlin, Germany
Marie-France Sagot	INRIA, France
Cenk Sahinalp	Simon Fraisher University, Canada
David Sankoff	University of Ottawa, Canada
Russell Schwartz	Carnegie Mellon University, USA
Eran Segal	Weizmann Institute, Israel
Roded Sharan	Tel Aviv University, Israel
Mona Singh	Princeton University, USA
Donna Slonim	Tufts University, USA
Terry Speed	UC Berkeley, USA
Peter Stadler	Universität Leipzig, Germany
Anna Tramontano	University of Rome, Italy
Alfonso Valencia	Spanish National Cancer Research Centre, Spain
Ana Tereza Vasconcelos	LNCC, Brazil
Martin Vingron	Max Planck Institute for Molecular Genetics, Germany
Jerome Waldispuhl	McGill University, Canada
Tandy Warnow	University of Texas, Austin, USA
Eric Xing	Carnegie Mellon University, USA
Jinbo Xu	Toyota Technology Institute Chicago, USA

## Steering Committee

Serafim Batzoglou	Stanford University, USA
Bonnie Berger	MIT, USA
Sorin Istrail	Brown University, USA
Thomas Lengauer	Max Planck Institute for Informatics, Germany
Michal Linial	The Hebrew University of Jerusalem, Israel
Pavel Pevzner	University of California, San Diego, USA
Terry Speed	University of California, Berkeley, USA
Martin Vingron (Chair)	Max Planck Institute for Molecular Genetics, Germany

## Organizing Committee

João Carriço	University of Lisbon, Portugal
Paulo Fonseca	INESC-ID / IST, Portugal
Ana Teresa Freitas	INESC-ID / IST, Portugal
José Pereira Leal	Instituto Gulbenkian da Ciencia, Portugal
Sara Madeira	INESC-ID / IST, Portugal
Arlindo Oliveira (Chair)	INESC-ID / IST, Portugal
Luís Rocha	Indiana University, Bloomington, USA
Sara Silva	INESC-ID, Portugal
Susana Vinga	INESC-ID / IST, Portugal

## Previous RECOMB Meetings

Dates	Hosting Institution	Program Chair	Conference Chair
January 20-23, 1997, Santa Fe, NM, USA	Sandia National Lab	Michael Waterman	Sorin Istrail
March 22-25, 1998 New York, NY, USA	Mt. Sinai School of Medicine	Pavel Pevzner	Gary Benson
April 22-25, 1999 Lyon, France	INRIA	Sorin Istrail	Mireille Regnier
April 8-11, 2000 Tokyo, Japan	University of Tokyo	Ron Shamir	Satoru Miyano
April 22-25, 2001 Montreal, Canada	Université de Montreal	Thomas Lengauer	David Sankoff
April 18-21, 2002 Washington, DC, USA	Celera	Gene Myers	Sridhar Hannenhalli
April 10-13, 2003 Berlin, Germany	German Federal Ministry for Education and Research	Webb Miller	Martin Vingron
March 27-31, 2004 San Diego, USA	University of California, San Diego	Dan Gusfield	Philip E. Bourne
May 14-18, 2005 Boston, MA, USA	Broad Institute of MIT and Harvard	Satoru Miyano	Jill P. Mesirov and Simon Kasif
April 2-5, 2006 Venice, Italy	University of Padova	Alberto Apostolico	Concettina Guerra
April 21-25, 2007 San Francisco, CA, USA	QB3	Terry Speed	Sandrine Dudoit
March 30-April 2, 2008 Singapore, Singapore	National University of Singapore	Martin Vingron	Limsoon Wong
May 18-21, 2009 Tucson, AZ, USA	University of Arizona	Serafim Batzoglou	John Kececioglu

## External Reviewers

Akavia, Uri David	Bilmes, Jeff
Alkan, Can	Brudno, Michael
Alterovitz, Gil	Buhler, Jeremy
Andreotti, Sandro	Cannistraci, Carlo
Anton, Brian	Carlbach, Eyal
AnunciaÃo, Orlando	Carvalho, Alexandra
Arndt, Peter	Chen, Bo-juen
Arvestad, Lars	Chuang, Han-Yu
Atias, Nir	Chung, Ho-Ryun
Aydin, Zafer	Codeco, Claudia
Bailey, Timothy L.	Dao, Phuong
Bandyopadhyay, Sourav	Darling, Aaron
Banks, Eric	David Amir, El-ad
Baran, Yael	David, Matei
Ben-Hur, Asa	DeConde, Robert
Benyamini, Tomer	Denise, Alain
Bertsch, Andreas	Domingues, Francisco
Bielow, Chris	Donald, Bruce Randall

Dotan, Dikla  
 Efros, Tal  
 Farahani, Hossein  
 Fonseca, Paulo  
 Fox, Andrew  
 Francisco, Alexandre  
 Frank, Ari  
 Fromer, Menachem  
 Fujita, Andre  
 Furlotte, Nicholas  
 Gitter, Anthony  
 Goeke, Jonathan  
 Gottlieb, Assaf  
 Gusfield, Dan  
 Haas, Stefan  
 Hach, Faraz  
 Hackermueller, Joerg  
 Hajiresouliha, Iman  
 Han, Buhm  
 Hannum, Gregory  
 He, Dan  
 Hescott, Ben  
 Hibbs, Matthew  
 Hoffman, Michael  
 Hofree, Matan  
 Homer, Nils  
 Hormozdiari, Fereydoun  
 Huang, Yang  
 Imoto, Seiya  
 Jaitly, Navdeep  
 Jeffery, Ian  
 Kang, Eun Yong  
 Kang, Hyun Min  
 Kapp, Eugene  
 Karakoc, Emre  
 Kehr, Birte  
 Khan, Zia  
 Khardon, Roni  
 Kim, Seyoung  
 Kirkpatrick, Bonnie  
 Klau, Gunnar W.  
 Klitford, Niels  
 Kohlbacher, Oliver  
 Kundaje, Anshul  
 Kuo, Dwight

Kuser, Paula  
 Langlois, Robert  
 Lasserre, Julia  
 Lee, Byoungkoo  
 Lee, Seunghak  
 Lengauer, Thomas  
 Lin, Yu  
 Litvin, Oren  
 Liu, Manway  
 Lonardi, Stefano  
 Lu, Hui  
 Ma, Jian  
 Magger, Oded  
 Mahony, Shaun  
 Manke, Thomas  
 Mann, Tobias  
 McIlwain, Sean  
 McLoughlin, Kevin  
 Medvedev, Paul  
 Melamed, Rachel  
 Mendes, Nuno  
 Miklos, Istvan  
 Misra, Navodit  
 Molla, Michael  
 Mongiovi, Misael  
 Nachman, Iftach  
 Nagasaki, Masao  
 Navarro, Gonzalo  
 Naxerova, Kamila  
 Neuburger, Daniel  
 Niida, Atsushi  
 Noboru, Sakabe  
 Noto, Keith  
 O'Rourke, Sean  
 Oliveira, Sara  
 Ortega, Miguel  
 Pasaniuc, Bogdan  
 Pe'er, Itsik  
 Pham, Son  
 Pisanti, Nadia  
 Ponty, Yann  
 Pop, Mihai  
 Puniyani, Kriti  
 Purdom, Elizabeth  
 Pushkarev, Dmitry

Qi, Yanjun  
Qi, Yuan  
Rajan, Vaibhav  
Ramoni, Marco  
Raskutti, Garvesh  
Reza-Chitsaz, Hamid  
Rheinbau, Esther  
Rodriguez, Jesse  
Rolfe, Alex  
Rosset, Saharon  
Russo, LuÃs  
Salari, Rahele  
Sanchez Garcia, Felix  
Sankararaman, Sriram  
Sargeant, Toby  
Schaub, Marc  
Schoenhuth, Alexander  
Serin, Akdes  
Sharon, Itai  
Shibuya, Tetsuo  
Shimamura, Teppei  
Shkolnisky, Yoel  
Shomron, Noam  
Shringarpure, Suyash  
Sievers, Fabian  
Singer, Amit  
Singh, Rohit  
Sommer, Ingolf  
Song, Le  
Sridhar, Srinath

Steel, Michael  
Steffen, Martin  
Swart, Estienne  
Swenson, Krister  
Szczyrek, Ewa  
Tamada, Yoshinori  
Tannier, Eric  
Tompa, Martin  
Tsai, Ming-Chi  
Tuller, Tamir  
Turcan, Sevin  
Turcotte, Marcel  
Ulitsky, Igor  
Uyar, Bora  
Veiga, Diogo  
Vernot, Benjamin  
Vilela, Marco  
Wall, Dennis  
Wei, Xintao  
Wilm, Andreas  
Xu, Andrew Wei  
Xuan, Jianhua  
Yamaguchi, Rui  
Yang, Shu  
Yanover, Chen  
Ye, Jieping  
Zaitlen, Noah  
Zemojtel, Tomasz  
Zhang, Xiuwei  
Zhou, Xianghong (Jasmine)

# Table of Contents

An Algorithmic Framework for Predicting Side-Effects of Drugs . . . . .	1
<i>Nir Atias and Roded Sharan</i>	
SubMAP: Aligning Metabolic Pathways with Subnetwork Mappings . . . .	15
<i>Ferhat Ay and Tamer Kahveci</i>	
Admixture Aberration Analysis: Application to Mapping in Admixed Population Using Pooled DNA . . . . .	31
<i>Sivan Bercovici and Dan Geiger</i>	
Pathway-Based Functional Analysis of Metagenomes . . . . .	50
<i>Sivan Bercovici, Itai Sharon, Ron Y. Pinter, and Tomer Shlomi</i>	
Hierarchical Generative Biclustering for MicroRNA Expression Analysis . . . . .	65
<i>José Caldas and Samuel Kaski</i>	
Subnetwork State Functions Define Dysregulated Subnetworks in Cancer . . . . .	80
<i>Salim A. Chowdhury, Rod K. Nibbe, Mark R. Chance, and Mehmet Koyutürk</i>	
Proteome Coverage Prediction for Integrated Proteomics Datasets . . . .	96
<i>Manfred Claassen, Ruedi Aebersold, and Joachim M. Buhmann</i>	
Discovering Regulatory Overlapping RNA Transcripts . . . . .	110
<i>Timothy Danford, Robin Dowell, Sudeep Agarwala, Paula Grisafi, Gerald Fink, and David Gifford</i>	
Alignment-Free Phylogenetic Reconstruction . . . . .	123
<i>Constantinos Daskalakis and Sebastien Roch</i>	
Inference of Isoforms from Short Sequence Reads (Extended Abstract) . . . . .	138
<i>Jianxing Feng, Wei Li, and Tao Jiang</i>	
The Clark Phase-able Sample Size Problem: Long-Range Phasing and Loss of Heterozygosity in GWAS . . . . .	158
<i>Bjarni V. Halldórsson, Derek Aguiar, Ryan Tarpine, and Sorin Istrail</i>	
A New Algorithm for Improving the Resolution of Cryo-EM Density Maps . . . . .	174
<i>Michael Hirsch, Bernhard Schölkopf, and Michael Habeck</i>	

Towards Automated Structure-Based NMR Resonance Assignment . . . . .	189
<i>Richard Jang, Xin Gao, and Ming Li</i>	
Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra . . . . .	208
<i>Kyowon Jeong, Sangtae Kim, Nuno Bandeira, and Pavel A. Pevzner</i>	
naiveBayesCall: An Efficient Model-Based Base-Calling Algorithm for High-Throughput Sequencing . . . . .	233
<i>Wei-Chun Kao and Yun S. Song</i>	
Extracting Between-Pathway Models from E-MAP Interactions Using Expected Graph Compression . . . . .	248
<i>David R. Kelley and Carl Kingsford</i>	
Simultaneous Identification of Causal Genes and Dys-Regulated Pathways in Complex Diseases . . . . .	263
<i>Yoo-Ah Kim, Stefan Wuchty, and Teresa M. Przytycka</i>	
Incremental Signaling Pathway Modeling by Data Integration . . . . .	281
<i>Geoffrey Koh, David Hsu, and P.S. Thiagarajan</i>	
The Poisson Margin Test for Normalisation Free Significance Analysis of NGS Data . . . . .	297
<i>Adam Kowalczyk, Justin Bedo, Thomas Conway, and Bryan Beresford-Smith</i>	
Compressing Genomic Sequence Fragments Using SLIMGENE . . . . .	310
<i>Christos Kozanitis, Chris Saunders, Semyon Kruglyak, Vineet Bafna, and George Varghese</i>	
On the Genealogy of Asexual Diploids . . . . .	325
<i>Fumei Lam, Charles H. Langley, and Yun S. Song</i>	
Genovo: <i>De Novo</i> Assembly for Metagenomes . . . . .	341
<i>Jonathan Laserson, Vladimir Jojic, and Daphne Koller</i>	
MoGUL: Detecting Common Insertions and Deletions in a Population . . . . .	357
<i>Seunghak Lee, Eric Xing, and Michael Brudno</i>	
Generalized Buneman Pruning for Inferring the Most Parsimonious Multi-state Phylogeny . . . . .	369
<i>Navodit Misra, Guy Belloch, R. Ravi, and Russell Schwartz</i>	
Seed Design Framework for Mapping SOLiD Reads . . . . .	384
<i>Laurent Noé, Marta Gîrdea, and Gregory Kucherov</i>	
Accurate Estimation of Expression Levels of Homologous Genes in RNA-seq Experiments . . . . .	397
<i>Bogdan Paşaniuc, Noah Zaitlen, and Eran Halperin</i>	



Cactus Graphs for Genome Comparisons . . . . .	410
<i>Benedict Paten, Mark Diekhans, Dent Earl, John St. John, Jian Ma, Bernard Suh, and David Haussler</i>	
IDBA - A Practical Iterative de Bruijn Graph De Novo Assembler . . . . .	426
<i>Yu Peng, Henry C.M. Leung, S.M. Yiu, and Francis Y.L. Chin</i>	
Predicting Nucleosome Positioning Using Multiple Evidence Tracks . . . . .	441
<i>Sheila M. Reynolds, Zhiping Weng, Jeff A. Bilmes, and William Stafford Noble</i>	
Dense Subgraphs with Restrictions and Applications to Gene Annotation Graphs . . . . .	456
<i>Barna Saha, Allison Hoch, Samir Khuller, Louiqa Raschid, and Xiao-Ning Zhang</i>	
Time and Space Efficient RNA-RNA Interaction Prediction via Sparse Folding . . . . .	473
<i>Raheleh Salari, Mathias Möhl, Sebastian Will, S. Cenk Sahinalp, and Rolf Backofen</i>	
HLA Type Inference via Haplotypes Identical by Descent . . . . .	491
<i>Manu N. Setty, Alexander Gusev, and Itsik Pe'er</i>	
Algorithms for Detecting Significantly Mutated Pathways in Cancer . . . . .	506
<i>Fabio Vandin, Eli Upfal, and Benjamin J. Raphael</i>	
Leveraging Sequence Classification by Taxonomy-Based Multitask Learning . . . . .	522
<i>Christian Widmer, Jose Leiva, Yasemin Altun, and Gunnar Rätsch</i>	
A Novel Abundance-Based Algorithm for Binning Metagenomic Sequences Using $l$ -Tuples . . . . .	535
<i>Yu-Wei Wu and Yuzhen Ye</i>	
A Markov Random Field Framework for Protein Side-Chain Resonance Assignment . . . . .	550
<i>Jianyang Zeng, Pei Zhou, and Bruce Randall Donald</i>	
Genomic DNA $k$ -mer Spectra: Models and Modalities (Abstract) . . . . .	571
<i>Benny Chor, David Horn, Nick Goldman, Yaron Levy, and Tim Massingham</i>	
Deciphering the Swine-Flu Pandemics of 1918 and 2009 (Abstract) . . . . .	572
<i>Richard Goldstein, Mario dos Reis, Asif Tamuri, and Alan Hay</i>	
Distinguishing Direct versus Indirect Transcription Factor-DNA Interactions (Abstract) . . . . .	574
<i>Raluca Gordân, Alexander J. Hartemink, and Martha L. Bulyk</i>	

A Self-regulatory System of Interlinked Signaling Feedback Loops Controls Mouse Limb Patterning (Abstract) . . . . .	575
<i>Jean-Denis Benazet, Mirko Bischofberger, Eva Tiecke, Alexandre Goncalves, James F. Martin, Aime Zuniga, Felix Naef, and Rolf Zeller</i>	
Automated High-Dimensional Flow Cytometric Data Analysis (Abstract) . . . . .	577
<i>Saamyadipta Pyne, Xinli Hu, Kui Wang, Elizabeth Rossin, Tsung-I Lin, Lisa Maier, Clare Baecher-Allan, Geoffrey McLachlan, Pablo Tamayo, David Hafler, Philip De Jager, and Jill Mesirov</i>	
Discovering Transcriptional Modules by Combined Analysis of Expression Profiles and Regulatory Sequences (Abstract) . . . . .	578
<i>Yonit Halperin, Chaim Linhart, Igor Ulitsky, and Ron Shamir</i>	
<b>Author Index</b> . . . . .	581

# An Algorithmic Framework for Predicting Side-Effects of Drugs

Nir Atias and Roded Sharan

Blavatnik School of Computer Science  
Tel Aviv University  
Tel Aviv 69978, Israel  
{atiasnir,roded}@post.tau.ac.il

**Abstract.** One of the critical stages in drug development is the identification of potential side effects for promising drug leads. Large scale clinical experiments aimed at discovering such side effects are very costly and may miss subtle or rare side effects. To date, and to the best of our knowledge, no computational approach was suggested to systematically tackle this challenge. In this work we report on a novel approach to predict the side effects of a given drug. Starting from a query drug, a combination of canonical correlation analysis and network-based diffusion are applied to predict its side effects.

We evaluate our method by measuring its performance in cross validation using a comprehensive data set of 692 drugs and their known side effects derived from package inserts. For 34% of the drugs the top scoring side effect matches a known side effect of the drug. Remarkably, even on unseen data, our method is able to infer side effects that highly match existing knowledge. Our method thus represents a promising first step toward shortcutting the process and reducing the cost of side effect elucidation.

**Keywords:** Prediction, Canonical correclation analysis, Network diffusion, Drug targets.

## 1 Introduction

Systems medicine is an emerging discipline in systems biology that aims at integrating clinical databases with large scale molecular interaction data to elucidate diseases and drugs [1]. Applications of such approaches range from predicting gene-disease associations and drug-target relations [2] to discovering new drugs [1].

Beyond the development of new drug leads, a critical stage in drug development is the identification of side effects that result from treatment with the drug. Drug safety has gained much attention in recent years, and has become a serious bottleneck in drug development, leading to the reduction in the number of newly approved drugs despite the enormous research efforts invested in drug discovery [3]. The elucidation of adverse reactions may occur long after the approval of a drug, as in the case of rosiglitazone maleate (Avienda <sup>®</sup>), and

can even lead to discontinuing the use of the drug, as in the case of rofecoxib (Vioxx <sup>®</sup>) (see also [4]).

The only attempt we are aware of to predict side effects is due to Xie et al. [5]. They used protein-ligand binding predictions to identify off-targets for a given drug. The latter were used to pinpoint known pathways that are likely to be affected by the drug and consequently predict its side effects. This approach depends on protein structure information and accurate pathway information, which greatly limits its applicability. In particular, biological processes involved in side effect reaction to treatment are still largely unknown and inferring side effects, even when given the respective drug targets, remains a formidable task [6].

In contrast to the sparse work on side effect prediction, the related area of elucidating gene-disease and drug-target associations has become very active in recent years. State of the art methods for predicting gene-disease associations are based on the observation that genes that cause similar diseases tend to lie close to one another in a network of protein-protein interactions [7,8]. Given a query disease, genes causing similar diseases are identified, and a network-based computation is used to prioritize candidate genes according to their proximity to this initial set [9,10,11]. Several methods have been suggested for drug-target prediction. Campillos et al. [2] construct a comprehensive drug-side effect data set and use it, in conjunction with chemical properties, to define a similarity metric between drugs. Given a query drug, they identify similar drugs and propose their targets as candidate targets for the drug. Yildirim et al. [12] examine a drug-target network in which drugs are connected based on shared targets and find that drug cluster according to the Anatomical Therapeutic Chemical (ATC) classification. Despite the insights offered by this network, no prediction scheme was suggested. A somewhat related work by Yang et al. [13] uses text mining to highlight genes responsible for serious adverse drug reactions. Finally, Kutalik et al. [14] integrate gene expression data and drug response data under different cell lines. They identify co-modules of genes and drugs with similar behavior across a subset of the cell lines, leading to the prediction of new drug targets.

Here we present a first systematic approach for predicting side effects for drugs. Our approach combines two algorithms to predict side effects. The first algorithm is based on canonical correlation analysis which is used to obtain a low dimensional subspace that jointly contains drug-side effect associations and molecular data on drugs, such as their chemical structure. Data on new drug queries are projected onto this subspace and an efficient algorithm is used to identify corresponding side effect vectors that best correlate with the projected data. The second algorithm is based on diffusion in a side effect similarity network. Starting from a prior solution that is based on the side effects of drugs that are similar to the query, a diffusion process is used to obtain final scores that are smooth over the network.

We evaluate our method by measuring its performance in 20-fold cross validation using a comprehensive data set of 692 drugs and their known side effects derived from package inserts. For 34% of the drugs the top scoring side effect matches a known side effect of the drug; for almost two thirds of the drugs our

method infers a correct side effect among the five top ranking predictions. In comparison, applying the algorithm to randomized instances, “correct” predictions are obtained for only 10% (top ranking) or 32% (among the five top ranking) of the drugs. We further validate our method in a blind test on  $\sim 450$  drugs that were not part of the initial data, but for which some side effect information exists in the literature. Remarkably, even on these unseen data, our method is able to infer side effects that highly match existing knowledge: for 45% of the drugs, a correct side effect is included among the five top ranking predictions. Finally, we show the utility of our method in drug target elucidation. We make predictions for over 4,000 drugs for which no side effect information is readily available. We then show a significant correlation between the side effect similarity and target similarity among these drugs. Not only does this agree with a previous study that used this correlation to predict drug targets [2], but importantly it suggests that target prediction algorithms can be applied also in the vast regime of drugs whose side effects have not been mapped to date.

## 2 Algorithmic Approach

We present two novel algorithms for predicting side effects, which are then combined to yield the final ranking of side effects for a given drug. The first algorithm is based on canonical correlation analysis. It requires as input an attribute matrix describing the drugs. In a training phase it learns a linear projection of the attribute and side effect data onto a joint low-dimensional space such that per drug, the correlation between the projected vectors of attributes and side effects is maximized. This projection is then used to infer the side effects of a test drug. The second algorithm is based on diffusion in a side effect similarity network. Given a query drug, the algorithm first identifies side effects of similar drugs. Starting from these side effects, a diffusion process is executed to obtain a final ranking that is smooth over the side effect network.

In the following we denote the number of drugs by  $n$  and the number of side effects by  $m$ . We assume that we are given as input a drug attribute matrix  $R_{p \times n}$ , in which each drug is described by a set of  $p$  attributes; a drug-side effect association matrix  $E_{m \times n}$ ; and an attribute vector  $q$  for a query drug. In a preprocessing step we normalize the rows of  $E$  and  $R$  to have mean 0.

### 2.1 Canonical Correlation Analysis

In canonical correlation analysis we aim to uncover and exploit the correlation between the two data sets that represent the drugs,  $R$  and  $E$  in our case, by projecting these data sets into a joint space and using the projection for the prediction task. We assume that corresponding vectors in each of the data sets should be highly correlated under some joint representation. Intuitively, our objective is to find two projection matrices,  $(W_E)_{m \times k}$  and  $(W_R)_{p \times k}$ , that project  $E$  and  $R$  onto a common  $k$ -dimensional subspace in which the correlations between projected vectors corresponding to the same drugs are maximized. The

projection vectors are chosen so that the set of projected vectors under each of the data sets will be orthonormal.

Formally, the problem is defined as follows:

$$\begin{aligned} \max_{W_E, W_R} \operatorname{Tr}(W_E^T E R^T W_R), \quad \text{subject to} \\ W_E^T E E^T W_E = W_R^T R R^T W_R = I \end{aligned} \quad (1)$$

where  $\operatorname{Tr}(M)$  is the trace of  $M$ . As shown in the Appendix, the resulting optimization problem can be solved by reducing it into an eigenvector problem on an appropriately defined matrix, and using the  $k$  eigenvectors with the largest eigenvalues to define the projection.

To avoid over-fitting and to account for numerical instabilities we use a regularized version of CCA [15]. The regularization takes additional regularization factors  $\eta_E$  and  $\eta_R$  which are used to penalize the norm of the column vectors of  $W_E$  and  $W_R$ . Instead of using two regularization factors we follow Wolf et al. [16] and use a single additional regularization parameter,  $\eta$ , and the largest eigenvalues,  $\lambda_E$  and  $\lambda_R$ , of  $E E^T$  and  $R R^T$ , respectively (see Appendix).

Finally, we use the projection matrices to compute a score vector for the query drug. To this end, the attribute vector  $q$  of the query drug is projected onto the subspace identified by the CCA:  $q_{proj} = W_R^T \cdot q$ . In accordance with the goal of CCA, we seek a corresponding side effect vector  $v$  whose projection maximizes the correlation to  $q_{proj}$ . Formally, we seek:

$$\max_v \frac{q_{proj}^T W_E^T v}{|q_{proj}| \|W_E^T v\|} \quad (2)$$

The maximum is achieved when  $W_E^T v = q_{proj}$ ; however, as  $W_E^T$  projects  $v$  into a smaller subspace, the system of equations is under-determined. To obtain a unique solution,  $f$ , we use the pseudoinverse of  $W_E^T$ , denoted by  $(W_E^T)^\dagger$ . In general, a pseudoinverse is computed using singular value decomposition, but here we can use the specific structure of  $W_E$  to compute it more efficiently using matrix multiplication, as detailed in the Appendix.

## 2.2 Diffusion-Based Prediction

The second algorithm that we use is based on a diffusion process in a side effect similarity matrix, aiming to score side effects so that: (i) prior information is taken into account; and (ii) similar side effects receive similar scores. Such an approach was applied successfully for predicting disease-causing genes [10].

Formally, given a similarity matrix between side effects ( $S$ ) and a prior information vector  $y$ , we seek a score vector  $f$  which satisfies:

$$f = \alpha S \cdot f + (1 - \alpha) y \quad (3)$$

where  $\alpha \in [0, 1]$  is a parameter reflecting the relative importance of the two (possibly contradicting) requirements on  $f$ .

We build  $S$  based on  $E$ , by measuring the Jaccard coefficient between the sets of drugs associated with each side effect. Formally, let  $\Gamma(s)$  denote the set of drugs associated with side effect  $s$ . Then the similarity between side effects  $i$  and  $j$  is given by the Jaccard coefficient of their corresponding drug sets:

$$\tilde{S}_{i,j} = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}. \quad (4)$$

To account for the different similarity profiles of different side effects, we normalize the similarities by setting  $S_{i,j} = \tilde{S}_{i,j} / \sqrt{P_i \cdot P_j}$ , where  $P_i = \sum_j \tilde{S}_{i,j}$ .

The computation of the prior vector is based on a similarity function between drugs. The latter is computed using  $R$  and its specific definition depends on the attribute data at hand, as described in Section 3.1. Let  $D_{q,d}$  denote the similarity between the query drug  $q$  and any other drug  $d$ . We apply a nearest neighbor approach, defining the prior value for side effect  $s$  as the highest similarity score  $D_{q,d}$  between a drug  $d$  and the query, across all drugs associated with  $s$ :  $y_s = \max_{d \in \Gamma(s)} \{D_{q,d}\}$ .

In [17] it is shown that if the eigenvalues of  $S$  are in  $[-1, 1]$  (which is the case under our normalization) then  $f$  can be computed using an iterative process

$$f^0 = y; \quad f^t = \alpha S \cdot f^{t-1} + (1 - \alpha) y \quad (5)$$

which efficiently converges to the analytical solution:  $f = (I - \alpha S)^{-1} (1 - \alpha) y$ .

### 2.3 Merging Score Vectors

Invoking the CCA based prediction and the diffusion based prediction yields two score vectors. Different strategies for merging these two vectors into a single ranking can be applied. Merging the two score vectors directly is problematic as the scores are not necessarily comparable. We follow ideas from Lin et al. [18], who use a logistic function for the merging. The logistic function is a monotonic transformation of the score, thus preserving the relative ranking of each algorithm on the one hand, while rescaling the scores to the same range on the other hand.

Formally, given score vectors  $s_1$  and  $s_2$ , with mean values  $\bar{s}_1$  and  $\bar{s}_2$ , respectively, the combined score vector is given by:

$$\text{score}(s_1, s_2) = \frac{1}{2} \left( \frac{1}{1 + e^{-(s_1 - \bar{s}_1)}} + \frac{1}{1 + e^{-b - a(s_2 - \bar{s}_2)}} \right) \quad (6)$$

where  $a$  and  $b$  are two free parameters which adjust between the two scoring systems.

### 2.4 Parameter Tuning and Performance Evaluation

The prediction algorithm has several parameters. Two parameters are used by the CCA algorithm:  $\eta$  – the regularization parameter, and  $k$  – the dimension

of the subspace to which the data are projected. One parameter is used by the diffusion algorithm:  $\alpha$  – the relative weight of the prior term vs. the smoothing term. Two final parameters,  $a$  and  $b$ , control the merging of the two score vectors.

We tune the parameters using grid search in a cross validation setting. Specifically, in each iteration of a 20-fold cross validation, 5% of the drugs serve as a test set and their side effect associations are hidden; 5% additional drugs serve as an internal test set to tune the parameters; the rest 90% of the drugs are used for training. First, the parameters of the two algorithms,  $\eta$ ,  $k$  and  $\alpha$ , are learned, maximizing the performance of each algorithmic variant separately on the internal test data. Next, the mixing parameters  $a$  and  $b$  are learned. Finally, the learned parameters are used to evaluate the performance of the algorithm on the test data. We note that in each cross validation iteration, the CCA projection and the side effect similarity network are recomputed.

We measure the quality of the predictions by computing a precision-recall curve for varying numbers of predictions per drug. Given a desired number of predictions,  $k$ , we consider the union of the top  $k$  ranking predictions for all drugs and compute: (i) *precision* – the percent of correct predictions; and (ii) *recall* – the percent of true side effects that were recovered. To summarize the curve we compute the area under it, as well as the area under its leftmost section where the recall is smaller than 0.2. To resolve cases in which several side effects attain the same score, we adjust the ranks of these side effects to be their average (unadjusted) rank.

To assess the significance of the results obtained by the algorithm, we applied it also to randomized instances of the data. The randomization was performed by permuting the columns of the drug-side effect association matrix  $E$ , thus randomizing the relations between drugs and their side effect vectors, while preserving the distribution of side effects in the data.

## 3 Results

### 3.1 Data Retrieval and Similarity Computations

Drugs and their associated side effects were obtained from SIDER [19], an online database containing drug-side effect associations extracted from package inserts using text mining methods [2]. This data set spans 880 drugs, 1382 side effects, and 61,102 drug-side effect associations. Drugs and side effects vary greatly in their number of associations. Some effects are present in almost all drugs (e.g., dizziness, edema and nausea), while others are associated with very few drugs (e.g. flashbacks, rectal polyp); and similarly for drugs. Thus, we filtered from the association data drugs and side effects that lie at the top 10% (greater than 151 associations for drugs and 127 associations for side effects), as well as side effects and drugs having less than two association. The resulting drug-side effect network contained 692 drugs, 680 side effects and 12,871 associations. These data were represented in a binary association matrix,  $E$ , where  $E_{s,d} = 1$  if and only if drug  $d$  is associated with side effect  $s$ .



The prediction algorithm can be applied with various drug attribute schemes, drug similarity measures and side effect similarity measures. For drugs, we experimented with two supporting data sets: (i) chemical hashed fingerprints; and (ii) NCI-60 drug response data for the different drugs under different cell lines [14]. For side effects, we based our similarity computation on their sets of associated drugs (see Section 2).

*Chemical data based computation.* Structures for the drugs molecules were downloaded from PubChem [20]. Hashed fingerprints based on these chemical structures were computed using the open source Chemistry Development Kit (CDK) [21,22]. The description matrix,  $R$ , used by the CCA prediction algorithm, is the matrix whose columns are the hashed fingerprints.

The similarity score between drugs, used by the diffusion algorithm, was calculated according to the Tanimoto 2D score between the two fingerprints, which is equal to their Jaccard coefficient. Formally, let  $r^d$  denote the hashed fingerprint for drug  $d$  ( $r_i^d \in \{0, 1\}$ ,  $i \in 1 \dots 1024$ ). The similarity score between two drugs,  $j$  and  $l$ , is given by:

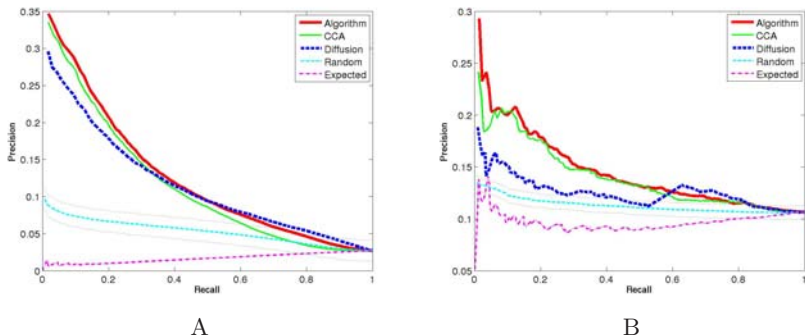
$$D_{j,l}^{(chem)} = \text{Tanimoto}(r^j, r^l) = \frac{\sum_i (r_i^j \cdot r_i^l)}{\sum_i (r_i^j + r_i^l - r_i^j \cdot r_i^l)} \quad (7)$$

*Response data based computation.* We downloaded the drug response data used in [14] from <http://serverdgm.unil.ch/bergmann/PingPong.html>. The data were used to build the description matrix  $R$ . An entry in  $R$  lists the concentration of a drug that is needed to achieve 50% growth inhibition under a certain cell line ( $\log(\text{GI}_{50})$ ). Missing data were replaced by the mean response to the drug over all cell lines. The similarity score between drugs, used by the diffusion algorithm, was calculated according to the Pearson correlation between the corresponding response profiles.

### 3.2 Chemical Structure Based Prediction Performance

In our first application of the algorithm we used the drug chemical structure information as supporting data. We tested the algorithm in a 20-fold cross validation setting, where in each cross validation iteration 5% of the data were hidden, serving as a test set, and the other 95% served as a training set. Within the training set, an internal cross validation was conducted to train the parameters of the algorithm as described in Section 2.4.

Overall, for 34.7% (240) of the 692 drugs the algorithm ranked first one of the known side effects of these drugs. For 63.4% (439) of the drugs, a correct side effect was ranked among the top five scoring side effects. In comparison, when applying our algorithm to randomized instances of these data, for only 68.1 ( $\pm 7.69$ , 9.85%) of the drugs, on average, the top ranking side effect matched a known side effect of the drug; and only 225.1 ( $\pm 12.8$ , 32.5%) of the drugs, on average, had a known side effect among the top five ranking side effects. These



**Fig. 1.** Performance evaluation. Dotted lines depict standard deviation for random curves. (A) Performance comparison using chemical structures as supporting data. (B) Performance comparison using drug response as supporting data.

**Table 1.** Performance statistics of the different algorithmic variants and a comparison to a random application. *Top1* lists the number of drugs having a known side effect ranked highest. *Top5* lists the number of drugs having at least one known side effect among the 5 highest ranking side effects. *Area* is the total area under the precision-recall curve; and *Area20* is the area under the leftmost (recall < 0.2) section of the precision-recall curve. The best result in each row appears in bold.

Data Set	Result	Combined alg.	CCA	Diffusion	Expected	Random
Chemical	Top1	<b>240</b>	232	206	0	68.16±7.69
	Top5	<b>439</b>	430	407	0	225.1±12.8
	Area	<b>0.1190</b>	0.1095	0.1111	0.0168	0.0524±0.0009
	Area20	<b>0.0483</b>	0.0465	0.0412	0.0017	0.0145±0.0005
Response	Top1	<b>17</b>	14	11	3	7.92±2.36
	Top5	<b>29</b>	26	25	23	24.86±3.23
	Area	<b>0.1419</b>	0.1382	0.1241	0.097	0.1122±0.005
	Area20	<b>0.0373</b>	0.035	0.0275	0.0204	0.0236±0.0024

marked differences are also reflected in the areas under the curve: 0.119 on the real data and 0.0524 ( $\pm 0.0009$ ) at random (see Figure 1A and Table 1).

We further compared the performance of the combined algorithm to those of applying the CCA or diffusion-based computations by themselves. As evident from the results in Figure 1A and Table 1, the combined algorithm outperforms the diffusion-based variant and is marginally better than the CCA based variant in all evaluation measures.

Some side effects are more prevalent than others and shared across many drugs. To examine the impact of side effect frequency on the prediction task, we have devised an algorithm that randomly ranks side effects according to their frequency distribution. The algorithm scores side effects by iteratively choosing side effects according to their empirical distribution in the training data, each

time incrementing their score. As shown in Figure 1A this algorithm performs worse than all other variants, suggesting that the prevalence of side effects is not sufficient to explain association with drugs.

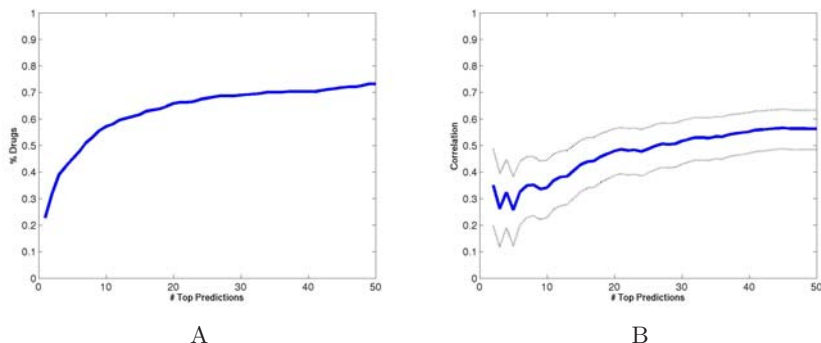
### 3.3 Response Based Prediction Performance

We additionally applied our algorithm using the drug response data. As the response information was not available for many of the drugs, the application was limited to 58 drugs, spanning 188 side effects. The algorithm ranked one of the known side effects highest for 17 (29%) of the drugs. For 29 (50%) drugs a correct side effect was ranked among the top 5 scoring side effects. These results significantly outperformed the random expectation (see Table 1). Precision-recall curves for the different algorithmic variants are displayed in Figure 1B. As for the chemical structure data, the combined algorithm outperformed diffusion based variant significantly and is marginally better than the CCA variant. The randomized algorithm based on side effect expectancy based on the occurrence distribution performs worse than all other variants.

### 3.4 A Large Scale Blind Test

To further validate our approach, we downloaded from DrugBank [23,24] a compilation of 4,335 drugs that were not available in SIDER. Chemical structures and hashed fingerprints for these new drugs were computed as described in section 3.1, and side effect rankings were calculated using the combined algorithm.

To evaluate the results of our prediction algorithm, we used the Hazardous Substances Data Bank (HSDB), an online peer reviewed database focusing on toxicology of potentially hazardous chemicals (see [25]). For 448 drugs that had matching records in HSDB, the text in the Human Health Effects section was downloaded and a simple textual search scheme was applied to extract annotated



**Fig. 2.** A blind test. (A) Percentage of new drugs with validated predictions in HSDB. (B) Correlation between number of validated predictions and the amount of information available for corresponding drugs. Dotted lines show the 95% confidence interval.

side effects. For 102 (22.8%) of the drugs, the side effect that was ranked highest by our algorithm was also associated to the corresponding drug in HSDB (see Figure 2A). For 201 (44.9%) of the drugs, one or more of the 5 top scoring side effects were confirmed by HSDB.

We believe that the accuracy in the validation is in fact higher, as only exact string matches were considered in the textual search and the side effect data are far from complete. To support this assertion, we calculated the correlation between the number of validated predictions and the length of the textual record in HSDB. For the 201 validated drugs mentioned above, we found a significant correlation between the quality of predictions and the amount of available information (Pearson  $r = 0.25$ ,  $p < 2.3e - 4$ ). The correlation increases as more predictions are taken into account (see Figure 2B).

### 3.5 Using Side Effect Predictions for Drug Target Elucidation

In a seminal paper, Campillos et al. [2] have shown that drugs with similar side effects are likely to share molecular targets. Exploiting this correlation they were able to predict new targets for drugs. However, their analysis was limited to drugs with known side effects. Our method has the potential to overcome this limitation as long as some molecular data is available on the drug in question.

To demonstrate the utility of our method in drug target elucidation, we applied it to predict the side effects of 4,335 drugs from DrugBank that do not have side effect information in SIDER. We then computed the correlation between two drug similarity matrices: one that is based on comparing the top  $k$  predicted side effects (via a Jaccard coefficient), and another that is based on comparing known drug targets (via a Jaccard coefficient). The Pearson correlation between the two similarity matrices varied for varying  $k$ , reaching a peak of 0.084 for  $k = 13$  (see Figure 3). This correlation was significantly higher than the random expectation (shuffling the drug-target associations while maintaining the same number of associated targets per drug). Expectedly, the correlation was lower than that observed for the drugs whose side effects are known (from SIDER).

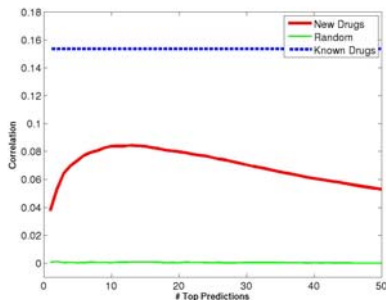


Fig. 3. Correlations of side-effect-based and target-based similarities

## 4 Conclusions

Our contribution in this paper is three fold: (i) We show that computational prediction of side effects of drugs is possible. We present an approach that combines correlation based analysis with network diffusion, achieving very high retrieval accuracy. In cross validation we are able to accurately predict side effects for up to two thirds of the drugs; in a blind test we are able to confirm our predictions for almost half of the drugs. (ii) We demonstrate the use of different data sets, such as chemical structure and cell line response, for the prediction task. The use of different data sets could potentially increase the sensitivity and specificity of the predictions. (iii) We find a significant correlation between the similarity of the predicted side effects of drugs and their targets, indicating the potential utility of our algorithm in drug target identification.

Several extensions of our work are possible. The CCA algorithm that we presented is limited to the analysis of one descriptive data set at a time. It is possible that using generalized canonical correlation analysis one could extend the method to take into account multiple data sets. The descriptive data used came from two sources: chemical structure information and cell line response data. Other sources of descriptive data could be used, most notably gene expression data in response to drug treatment such as those cataloged by the Connectivity Map project [1].

In summary, we believe that our algorithm constitutes a first step toward shortcutting the process of side effect identification in the development of new drugs.

## References

1. Lamb, J., Crawford, E., Peck, D., Modell, J., Blat, I., Wrobel, M., Lerner, J., Brunet, J., Subramanian, A., Ross, K., Reich, M., Hieronymus, H., Wei, G., Armstrong, S., Haggarty, S., Clemons, P., Wei, R., Carr, S., Lander, E., Golub, T.: The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313(5795), 1929–1935 (2006)
2. Campillos, M., Kuhn, M., Gavin, A., Jensen, L., Bork, P.: Drug target identification using side-effect similarity. *Science* 321(5886), 263–266 (2008)
3. Billingsley, M.: Druggable targets and targeted drugs: enhancing the development of new therapeutics. *Pharmacology* 82(4), 239–244 (2008)
4. Moore, T., Cohen, M., Furberg, C.: Serious adverse drug events reported to the food and drug administration, 1998–2005. *Arch. Intern. Med.* 167(16), 1752–1759 (2007)
5. Xie, L., Li, J., Bourne, P.: Drug discovery using chemical systems biology: identification of the protein-ligand binding network to explain the side effects of cetyl inhibitors. *PLoS Comput. Biol.* 5(5), e1000387 (2009)
6. Need, A., Motulsky, A., Goldstein, D.: Priorities and standards in pharmacogenetic research. *Nat. Genet.* 37(7), 671–681 (2005)
7. Oti, M., Snel, B., Huynen, M.A., Brunner, H.G.: Predicting disease genes using protein-protein interactions. *J. Med. Genet.* 43(8), 691–698 (2006)

8. Franke, L., van Bakel, H., Fokkens, L., de Jong, E.D., Egmont-Petersen, M., Wijmenga, C.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.* 78(6), 1011–1025 (2006)
9. Kohler, S., Bauer, S., Horn, D., Robinson, P.N.: Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.* 82(4), 949–958 (2008)
10. Vanunu, O., Sharan, R.: A propagation-based algorithm for inferring gene-disease associations. In: *German Conference on Bioinformatics*, pp. 54–52 (2008)
11. Wu, X., Jiang, R., Zhang, M.Q., Li, S.: Network-based global inference of human disease genes. *Mol. Syst. Biol.* 4, 189 (2008)
12. Yildirim, M.A., Goh, K.I., Cusick, M.E., Barabasi, A.L., Vidal, M.: Drug-target network. *Nat. Biotechnol.* 25(10), 1119–1126 (2007)
13. Yang, L., Xu, L., He, L.: A citationrank algorithm inheriting google technology designed to highlight genes responsible for serious adverse drug reaction. *Bioinformatics* 25(17), 2244–2250 (2009)
14. Kutalik, Z., Beckmann, J.S., Bergmann, S.: A modular approach for integrative analysis of large-scale gene-expression and drug-response data. *Nat. Biotechnol.* 26(5), 531–539 (2008)
15. Leurgans, S.E., Moyeed, R.A., Silverman, B.W.: Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society, Series B (Methodological)* 55(3), 725–740 (1993)
16. Wolf, L., Donner, Y.: An experimental study of employing visual appearance as a phenotype. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pp. 1–7 (2008)
17. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Scholkopf, B.: Learning with local and global consistency. In: *Advances in Neural Information Processing Systems*, vol. 16, pp. 321–328. MIT Press, Cambridge (2004)
18. Lin, W.-H., Hauptmann, A.: Merging rank lists from multiple sources in video classification. In: *Proc. IEEE International Conference on Multimedia and Expo ICME 2004*, vol. 3, pp. 1535–1538 (2004)
19. Kuhn, M., Campillos, M., Letunic, I., Jensen, L., Bork, P.: A side effect resource to capture phenotypic effects of drugs (submitted), <http://sideeffects.embl.de/>
20. Wheeler, D.L., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., Dicuccio, M., Edgar, R., Federhen, S., Feolo, M., Geer, L.Y., Helmberg, W., Kapustin, Y., Khovayko, O., Landsman, D., Lipman, D.J., Madden, T.L., Maglott, D.R., Miller, V., Ostell, J., Pruitt, K.D., Schuler, G.D., Shumway, M., Sequeira, E., Sherry, S.T., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusov, R.L., Tatusova, T.A., Wagner, L., Yaschenko, E.: Database resources of the national center for biotechnology information. *Nucleic Acids Res.* 36(Database issue), D13–D21 (2008)
21. Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., Willighagen, E.: The chemistry development kit (cdk): an open-source java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci.* 43(2), 493–500 (2003)
22. Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., Willighagen, E.L.: Recent developments of the chemistry development kit (cdk) - an open-source java library for chemo- and bioinformatics. *Curr. Pharm. Des.* 12(17), 2111–2120 (2006)
23. Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., Hassanali, M.: Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* 36(Database issue), D901–D906 (2008)

24. Wishart, D.S., Knox, C., Guo, A.C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., Woolsey, J.: Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res.* 34(Database issue), D668–D672 (2006)
25. Wexler, P.: Toxnet: an evolving web resource for toxicology and environmental health information. *Toxicology* 157(1-2), 3–10 (2001)

## Appendix

### Solving the CCA Optimization Problem

Given two descriptive matrices  $E$  and  $R$ , CCA aims at finding two projection matrices  $W_E$  and  $W_R$  so that the following correlation is maximized:

$$\max_{W_E, W_R} \text{Tr} (W_E^T E R^T W_R), \quad \text{subject to} \quad (8)$$

$$W_E^T E E^T W_E = W_R^T R R^T W_R = I$$

Denote  $C_{EE} = E E^T$ ,  $C_{ER} = E R^T$ ,  $C_{RE} = R E^T$  and  $C_{RR} = R R^T$ . Consider first the case where each of the projection matrices is a single vector, and define the following optimization problem:

$$\max_{w_e, w_r} \frac{w_e^T C_{ER} w_r}{\sqrt{w_e^T C_{EE} w_e \cdot w_r^T C_{RR} w_r}} \quad (9)$$

Since the expression to optimize is invariant under scaling of the projections  $w_e$  and  $w_r$ , one can fix the two terms in the denominator to 1 and optimize the numerator. The resulting Lagrangian is:

$$\mathcal{L}(\lambda_e, \lambda_r, w_e, w_r) = w_e^T C_{ER} w_r - \frac{\lambda_e}{2} (w_e^T C_{EE} w_e - 1) - \frac{\lambda_r}{2} (w_r^T C_{RR} w_r - 1)$$

Taking derivatives and comparing to zero we find that  $\lambda_e = \lambda_r = \lambda$  and, consequently, that  $w_r$  can be expressed as:

$$w_r = \frac{C_{RR}^{-1} C_{RE} w_e}{\lambda} \quad (10)$$

and that  $w_e$  is the solution to the generalized eigen problem:

$$C_{ER} C_{RR}^{-1} C_{RE} w_e = \lambda^2 C_{EE} w_e \quad (11)$$

Let  $W_R$  be the matrix whose columns are the vectors solving Eq. [10](#), and let  $W_E$  be the matrix whose columns are eigenvectors solving Eq. [11](#). Then

$$\begin{aligned} \text{Tr} (W_E^T C_{ER} W_R) &= \sum_{i=1}^k w_{e,i}^T C_{ER} w_{r,i} \\ &= \sum_{i=1}^k \frac{w_{e,i}^T C_{ER} C_{RR}^{-1} C_{RE} w_{e,i}}{\lambda_i} \\ &= \sum_{i=1}^k \frac{\lambda_i^2 w_{e,i}^T C_{EE} w_{e,i}}{\lambda_i} = \sum_{i=1}^k \lambda_i \end{aligned}$$

Thus choosing eigenvectors corresponding to the  $k$  largest eigenvalues will maximize the objective of Eq. [11](#).

It remains to show that this solution respects the optimization constraints. The constraints of the Lagrangian ensure that the entries along main diagonal of  $W_E^T E E^T W_E$  and  $W_R^T R R^T W_R$  are equal to one. To show that the off-diagonal elements of these matrices are zero, we apply the Cholesky decomposition to  $C_{EE}$  and  $C_{RR}$  (both are symmetric):  $C_{EE} = L_{EE} L_{EE}^T$  and  $C_{RR} = L_{RR} L_{RR}^T$ . Denoting  $u_e = L_{EE}^T w_e$  and  $A = L_{EE}^{-1} C_{ER} (L_{RR}^T)^{-1}$ , we can reformulate Eq. [11](#) as a standard eigen problem:

$$L_{EE}^{-1} C_{ER} (L_{RR}^T)^{-1} L_{RR}^{-1} C_{RE} (L_{EE}^T)^{-1} u_e = A A^T u_e = \lambda^2 u_e \quad (12)$$

As  $A A^T$  is symmetric, its eigenvectors  $u_e$  are orthogonal, implying that for  $i \neq j$ :  $w_{e,i}^T E E^T w_{e,j} = w_{e,i}^T L_{EE} L_{EE}^T w_{e,j} = u_{e,i}^T u_{e,j} = 0$ .

In the regularized version of CCA, the terms  $C_{EE}$  and  $C_{RR}$  in Eq. [9](#) are replaced with

$$\begin{aligned} C_{EE}^* &= (E E^T + \eta \lambda_E I) \\ C_{RR}^* &= (R R^T + \eta \lambda_R I) \end{aligned} \quad (13)$$

### Computing a Side Effect Vector with Highest Correlation

We wish to efficiently compute the vector  $f = (W_E^T)^{\dagger} q_{proj}$ . Using the notation above,  $u_e = L_{EE}^T w_e$ , and in matrix form,  $U_E = L_{EE}^T W_E$ . Substitute that into the equation above we get:

$$f = \left( (L_{EE}^T)^{-1} U_E \right)^{\dagger} q_{proj} \quad (14)$$

Since  $L_{EE}^T$  is invertible, the pseudoinverse of  $(L_{EE}^T)^{-1}$  is  $L_{EE}^T$ . Since  $U_E$  has linearly independent columns, its pseudoinverse is equal to  $(U_E^T U_E)^{-1} U_E^T$ . It follows that

$$\begin{aligned} f &= (U_E^T U_E)^{-1} U_E^T L_{EE}^T q_{proj} = U_E^T L_{EE}^T q_{proj} \\ &= L_{EE} U_{EE} q_{proj} = C_{EE} W_E q_{proj} \end{aligned}$$

Thus  $f$  can be computed using simple matrix multiplication.



# SubMAP: Aligning Metabolic Pathways with Subnetwork Mappings

Ferhat Ay and Tamer Kahveci

Computer and Information Science and Engineering,  
University of Florida, Gainesville, FL 32611  
fay@cise.ufl.edu, tamer@cise.ufl.edu

**Abstract.** We consider the problem of aligning two metabolic pathways. Unlike traditional approaches, we do not restrict the alignment to one-to-one mappings between the molecules of the input pathways. We follow the observation that in nature different organisms can perform the same or similar functions through different sets of reactions and molecules. The number and the topology of the molecules in these alternative sets often vary from one organism to another. In other words, given two metabolic pathways of arbitrary topology, we would like to find a mapping that maximizes the similarity between the molecule subsets of query pathways of size at most a given integer  $k$ . We transform this problem into an eigenvalue problem. The solution to this eigenvalue problem produces alternative mappings in the form of a weighted bipartite graph. We then convert this graph to a vertex weighted graph. The maximum weight independent subset of this new graph is the alignment that maximizes the alignment score while ensuring consistency. We call our algorithm **SubMAP** (**S**ubnetwork **M**appings in **A**lignment of **P**athways). We evaluate its accuracy and performance on real datasets. Our experiments demonstrate that SubMAP can identify biologically relevant mappings that are missed by traditional alignment methods and it is scalable for real size metabolic pathways.

**Availability:** Our software and source code in C++ is available at <http://bioinformatics.cise.ufl.edu/SubMAP.html>

## 1 Introduction

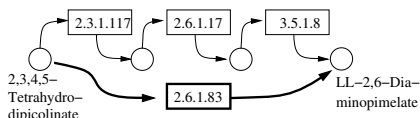
Biological pathways show how different molecules interact with each other to perform vital functions. In the literature, the terms “network” and “pathway” are used interchangeably for different types of interaction data. Metabolic pathways, an important class of biological pathways, represent how different compounds are transformed through various reactions. Analyzing these pathways is essential in understanding the machinery of living organisms.

The efforts on analyzing pathways can be classified into two types. The first type takes one pathway into account at a time and explores the important properties of that network such as its robustness [1], steady states [2] and modular structure [3]. The second type is the comparative approach which considers

multiple pathways to identify their frequent subgraphs [4,5] and their alignments [6,7,8,9,10,11]. Alignment is a fundamental type of comparative analysis which aims to identify similar parts between pathways. For metabolic pathways, these similarities provide insights for drug target identification [12,13], metabolic reconstruction of newly sequenced genome [14], phylogenetic reconstruction [15,16] and enzyme cluster and missing enzyme identification [17,18].

In the literature, alignment is often considered as finding one-to-one mappings of the molecules of two pathways. In this case, the global/local pathway alignment problems are GI/NP complete as the graph/subgraph isomorphism problems can be reduced to them in polynomial time [19]. A number of studies have been done to systematically align different types of biological networks. For metabolic pathways, Pinter *et al.* [6] devised an algorithm that aligns query pathways with specific topologies by using a graph theoretic approach. Tohsato *et al.* proposed two algorithms one relying solely on Enzyme Commission (EC [20]) numbers of enzymes and the other considering only the chemical structures of compounds of the query pathways [9,10]. Latterly, Cheng *et al.* developed a tool, *MetNetAligner*, for metabolic pathway alignment that allows a certain number of insertions and deletions of enzymes [11]. However, these methods do not integrate different types of information (e.g., topology, homology) and focus on a single similarity score (e.g., enzyme similarity, compound similarity, etc.). Furthermore, some of these methods limit the query pathways to certain topologies, such as trees, non-branching paths or limited cycles, which degrades their applicability to complex pathways. Recently, Singh *et al.* [21] and Ay *et al.* [7,8] combined both topological features and homological similarity of pairwise molecules to find the alignments of protein interaction networks and metabolic pathways respectively. These two algorithms showed that this integration increases the accuracy of alignment. Additionally, these methods do not restrict the topologies of query pathways and hence are applicable to arbitrarily complex pathways.

All the methods discussed above limit the possible molecule mappings to only one-to-one mappings. As also pointed out by Deutscher *et al.* [22] considering each molecule one by one fails to reveal its function(s) in complex pathways. This restriction prevents all the above methods from identifying biologically relevant mappings when different organisms perform the same function through varying number of steps. As an example, there are alternative paths for LL-2,6-Diaminopimelate production in different organisms [13,23]. Figure 1



**Fig. 1.** A portion of Lysine biosynthesis pathway. Each reaction is represented by the Enzyme Commission (EC) number of the enzyme that catalyze it. Each circle represents a compound. Humans use the path on the top with three reactions, whereas plants and Chlamydia can achieve this transformation directly by a single reaction through the path (**in bold**) at the bottom.

illustrates two paths both producing LL-2,6-Diaminopimelate starting from 2,3,4,5-Tetrahydrodipicolinate. The bottom path represents the shortcut used by Chlamydia and plants on the path to the synthesis of an important amino acid, L-Lysine. This shortcut is not available for humans since we lack LL-DAP aminotransferase (2.6.1.83). Humans use a three step process shown as the top path in Figure 1 to do this transformation. Thus, a meaningful alignment should match the top path with three reactions to the bottom with a single reaction when the human lysine biosynthesis pathway is aligned to the same pathway of a plant or Chlamydia. However, since these two paths have different number of reactions, traditional alignment methods fail to identify this mapping.

*Our aim in this paper is to design an algorithm that can accurately identify such biologically relevant mappings by allowing one-to-many mappings of molecules.* Note that, in Figure 1 the topologies of both reaction sets are linear paths. It is possible to have reaction sets with arbitrary topologies. Therefore, we use the term *subnetwork* to include all types of topologies. Also, since we only consider the sets of reactions that are connected, we will simply use the term *subnetwork* instead of *connected subnetwork*.

**Problem definition:** Here, we consider the problem of aligning two metabolic pathways. Unlike traditional alignment approach, we allow aligning a molecule of one pathway to a connected subnetwork of the other. More formally, let  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  be two query pathways and  $k$  be a positive integer. We want to find the mapping between the molecules of  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  with the largest alignment score, such that (1) each molecule in  $\mathcal{P}$  ( $\bar{\mathcal{P}}$ ) can map to a subnetwork of  $\bar{\mathcal{P}}$  ( $\mathcal{P}$ ) with at most  $k$  molecules and (2) each molecule can appear in at most one mapping.

The first condition above allows one-to-many mappings. The second condition enforces *consistency*. That is, if a molecule is already mapped alone or as a part of a subnetwork, it cannot map to another molecule. We elaborate on consistency and the problem definition later in Section 2. *Note that, allowing one-to-many mappings in alignment introduces new computational challenges that cannot be addressed using existing methods and hence novel methods are needed to tackle this problem.*

**Contributions:** In this paper, we propose a novel algorithm named **SubMAP** that finds subnetwork mappings in alignment of pathways. SubMAP accounts for both the effect of pairwise similarities (homology) and the organization of pathways (topology). This combination is motivated by its successful applications on pathway alignment by Singh *et al.* [21] and Ay *et al.* [78]. However, allowing one-to-many mappings makes it impossible to trivially extend these methods to our problem. To address this challenge, we map our problem to an eigenvalue problem. We solve this eigenvalue problem using an iterative technique called power method. The result of the power method converges to a principal eigenvector. This eigenvector defines a weighted bipartite graph where each node corresponds to a molecule or a subnetwork. The edges are only between two nodes from different pathways and their weights define the similarity of these nodes. Unlike the problem with only one-to-one molecule mappings, the resulting nodes of the bipartite graph can be intersecting as the same molecule can appear in more

than one subnetwork. We term such node pairs as *conflicting*. In order to ensure that the alignment is consistent, we construct a vertex weighted *conflict graph* with nodes representing a mapping of two subnetworks one from each pathway and edges representing a *conflict* between two mappings (i.e., they create inconsistency). The similarity values in the principal eigenvector are the weights of the nodes in conflict graph. Our algorithm aims to find the set of mappings (nodes) that has no conflicts (edges) and maximizes the total weight of nodes. This problem is equivalent to finding *maximum weight independent subset*. Since the maximum weight independent set problem is NP-hard, we use a heuristic to extract an independent set from the conflict graph which gives us a non-conflicting set of one-to-many mappings. We report these mappings as the *alignment* of the query pathways. *Our experiments on the metabolic pathways from KEGG [24] database suggest that SubMAP finds biologically meaningful alignments efficiently. Also, SubMAP is scalable as it aligns pathways with around 50 reactions while allowing subnetworks of size three in less than a minute.*

The rest of the paper is organized as follows. Section 2 describes our algorithm. Section 3 presents experimental results. Section 4 concludes the paper.

## 2 Our Algorithm: SubMAP

In this section, we present our algorithm for pairwise metabolic pathway alignment that allows one-to-many molecule mappings. We begin by introducing some notation that we use throughout this section. Then, we formally state the problem and describe the SubMAP algorithm in detail.

Let,  $\mathcal{P}$  be a pathway which is represented by a directed unweighed graph  $G = (V, E)$ . Here, we only use the reactions of the pathway in graph representation. Hence, the vertex set  $V = \{r_1, r_2, \dots, r_n\}$  is the set of all reactions of  $\mathcal{P}$ . We include a directed edge  $e_{ij}$  from  $r_i$  to  $r_j$  in  $E$  if and only if at least one output compound of  $r_i$  is an input compound of  $r_j$ . We call  $r_i$  a *backward neighbor* of  $r_j$  and  $r_j$  a *forward neighbor* of  $r_i$  if  $e_{ij} \in E$ . Note that reactions can be reversible (bi-directional) and hence both  $e_{ij}$  and  $e_{ji}$  can exist.

A *subnetwork* of a pathway is a subset of its reaction set such that the induced undirected graph of the elements of this subset forms a connected graph. Let  $R_i \subseteq V$  be such a subnetwork of  $\mathcal{P}$ . We define  $\mathcal{R}_k$  as  $\mathcal{R}_k = \{R_1, R_2, \dots, R_N\}$  where  $|R_i| \leq k$  for all  $i \in [1, N]$ . Here,  $|R_i|$  denotes the cardinality of the reaction set  $R_i$ . Verbally,  $\mathcal{R}_k$  is the set of all subnetworks of  $\mathcal{P}$  that have at most  $k$  reactions. Using this notation, we define a binary relation that maps a reaction of a query pathway to a subnetwork of the other as follows:

**Definition 1.** *Let  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  be two pathways and  $k$  be a positive integer. Also, let  $\mathcal{R}_k = \{R_1, R_2, \dots, R_N\}$  and  $\bar{\mathcal{R}}_k = \{\bar{R}_1, \bar{R}_2, \dots, \bar{R}_M\}$  be the sets of subnetworks with size at most  $k$  of  $\mathcal{P}$  and  $\bar{\mathcal{P}}$ . We define a binary relation between  $\mathcal{R}_k$  and  $\bar{\mathcal{R}}_k$  that allows one-to-many reaction mappings as  $\varphi : \varphi \subseteq (\mathcal{R}_1 \times \bar{\mathcal{R}}_k) \cup (\mathcal{R}_k \times \bar{\mathcal{R}}_1)$ .*

Clearly,  $\varphi$  allows one-to-one and one-to-many mappings. The cardinality of  $\varphi$  ( $|\varphi|$ ) is at most  $nM + mN - nm$  where  $n, m$  are the number of reactions of

$\mathcal{P}$  and  $\bar{\mathcal{P}}$  respectively. In other words, the number of all possible mappings is  $nM + mN - nm$ . The *alignment* of  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  is a binary relation that is a subset of all these possible mappings and satisfies certain criteria that we describe next.

Recall that for a mapping  $(R_i, \bar{R}_j) \in \varphi$  one of the  $R_i$  or  $\bar{R}_j$  can contain more than one reaction. Reporting this mapping as a part of our alignment implies that all the reactions of the subnetwork with multiple reactions are aligned to a single reaction of the other. To have a *consistent alignment* none of the reactions of these subnetworks can be included in any other mapping. Next, we formally define the term *conflict* to characterize this property.

**Definition 2.** Let  $\varphi$  be a binary relation and  $R_i, R_u \in \mathcal{R}_k$  and  $\bar{R}_j, \bar{R}_v \in \bar{\mathcal{R}}_k$ . The distinct pairs  $(R_i, \bar{R}_j) \in \varphi$  and  $(R_u, \bar{R}_v) \in \varphi$  **conflict** if and only if  $(R_i \cap R_u) \cup (\bar{R}_j \cap \bar{R}_v) \neq \emptyset$ .

Conflicts can cause inconsistencies about which reaction subset of one pathway should be aligned to the one of the other pathway. If  $\varphi$  has a conflicting pair of elements, we say  $\varphi$  is *inconsistent*. Since this is not a desirable property, we *limit our alignment to the consistent relations only*.

In order to find biologically relevant alignments we also need a meaningful scoring scheme. One standard scoring scheme for this purpose incorporates the homology of the aligned molecules with their topologies [7,8,21]. Here, we generalize this scheme to one-to-many mappings. We will elaborate on this similarity score later in Section 2.4. Next, we state our problem formally.

**Problem formulation:** Given  $k$  and two pathways  $\mathcal{P}$  and  $\bar{\mathcal{P}}$ , let  $\mathcal{R}_k$  and  $\bar{\mathcal{R}}_k$  be the sets of subnetworks with size at most  $k$  of  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  respectively. We want to find the *consistent* binary relation  $\varphi \subseteq (\mathcal{R}_1 \times \bar{\mathcal{R}}_k) \cup (\mathcal{R}_k \times \bar{\mathcal{R}}_1)$  that *maximizes* the summation of the similarity scores of the aligned subnetworks.

In the following, we present our algorithm SubMAP. Section 2.1 explains how we enumerate the subnetworks of query pathways. Section 2.2 and 2.3 discuss homological and topological similarities respectively. Section 2.4 describes the eigenvalue formulation and extraction of the alignment.

## 2.1 Enumeration of Connected Subnetworks

The first step of SubMAP is to create the sets of all connected subnetworks of size at most  $k$  for each query pathway. Here, we describe the enumeration process for a single query pathway. Let  $G = (V, E)$  represent a pathway and  $k$  be a positive integer. We construct the set of subnetworks  $\mathcal{R}_k$  as follows. For  $k = 1$   $\mathcal{R}_k = \mathcal{R}_1 = V$ . For  $k > 1$  we define  $\mathcal{R}_k$  recursively by using  $\mathcal{R}_{k-1}$ . At each recursive step we check for each reaction in  $V$  if it can be added to already enumerated subnetworks of size  $k - 1$  to create a new connected subnetwork of size  $k$ . This way the  $k$ th recursive step takes  $O(|V| \cdot (|\mathcal{R}_{k-1}| - |\mathcal{R}_{k-2}|))$  time.

The size of the set  $\mathcal{R}_k$  can be exponential in  $k$  when  $G$  is dense. However, metabolic pathways are usually sparse (on the average there are 2.5 forward neighbors per reaction). We observe that the number of subnetworks of real metabolic pathways for  $k = 3$  is around  $5|V|$  and for  $k = 4$  it is  $10|V|$  on the

average. In Section 3.2, we provide a detailed discussion of how  $|\mathcal{R}_k|$  changes with different pathway sizes and different  $k$  values.

## 2.2 Homological Similarity of Subnetworks

Recall that the relation  $\varphi$  maps a reaction to a subnetwork that can contain multiple reactions. This necessitates computing the similarity between reaction sets. Since reactions are defined by their input and output compounds (i.e., substrates and products) and the enzymes that catalyze them, we measure the homological similarity between reactions using the similarities of these components.

In the literature, there are alternative pairwise similarity scores for compounds, enzymes and reactions. Particularly, two well known measure are information content similarity for enzyme pairs [6] and SIMCOMP [25] for compound pairs. We denote these measures by  $SimE$  and  $SimC$  respectively. We defer the readers to Ay *et al.* [8] for details on computing these similarities. Here, we utilize these similarity measures to compute the homological similarity between two reaction sets. To calculate this, we first construct the sets of the unions of input compounds ( $I_i$ ), output compounds ( $O_i$ ) and enzymes ( $E_i$ ) of the reactions in each subnetwork  $R_i$ . For instance, in Figure 1 if we take upper path as the subnetwork  $R_i$ , then  $E_i = \{2.3.1.117, 2.6.1.17, 3.5.1.8\}$ . Let  $\gamma_e, \gamma_i, \gamma_o$  denote the relative weights of the similarities of enzymes, input compounds and output compounds respectively. We define  $SimRSet$  as:

$$SimRSet(R_i, \bar{R}_j) = \gamma_e W(E_i, \bar{E}_j, SimE) + \gamma_i W(I_i, \bar{I}_j, SimC) + \gamma_o W(O_i, \bar{O}_j, SimC)$$

Here  $W$  denotes the sum of edge weights of the pairs returned by the maximum weight bipartite matching (MWBM) of the two sets. MWBM finds an assignment between the nodes of two sets such that maximizes the sum of the weights of these assignments specified by the similarity score. We use  $\gamma_i = \gamma_o = 0.3$  and  $\gamma_e = 0.4$  as they provide a good balance between enzymes and compounds.

We calculate  $SimRSet$  for all possible one-to-many mappings between the subnetworks of two pathways. The number of possible pairings is  $nM + mN - nm$  where  $n, m$  are the number of reactions of  $\mathcal{P}, \bar{\mathcal{P}}$  and  $N = |\mathcal{R}_k|$  and  $M = |\bar{\mathcal{R}}_k|$ . Therefore, in this step, we calculate  $SimRSet$  function  $nM + mN - nm$  times. This way, we assess the homological similarities between all possible subnetwork mappings. Even though this scoring is a good measure of similarity, relying solely on this score ignores the topology similarity which we explain next.

## 2.3 Topological Similarity of Subnetworks

The motivation for utilizing topological similarity is that the induced topologies of two aligned subnetworks should also be similar. In other words, if  $R_i$  is mapped to  $\bar{R}_j$ , then their neighbors in the corresponding pathways should also be similar. Motivated by this, we first extend the neighborhood definition of reactions to reaction subnetworks. Then, we introduce the notion of *support* between two mappings.

**Definition 3.** Let  $R_i, R_u \in \mathcal{R}_k$ . Then,  $R_u$  is a **forward neighbor** of  $R_i$  ( $R_u \in FN(R_i)$ ) if and only if there exists  $r_a \in R_i$  and  $r_b \in R_u$  such that  $r_b$  is a forward neighbor of  $r_a$  or  $R_i \cap R_u \neq \emptyset$ .  $R_i$  is a **backward neighbor** of  $R_u$  ( $R_i \in BN(R_u)$ ) if and only if  $R_u$  is a forward neighbor of  $R_i$ .

**Definition 4.** Let  $R_i, R_u \in \mathcal{R}_k$  and  $\bar{R}_j, \bar{R}_v \in \bar{\mathcal{R}}_k$ . The mapping  $(R_i, \bar{R}_j)$  **supports** the mapping  $(R_u, \bar{R}_v)$  if and only if both  $R_j \in FN(R_i)$  and  $\bar{R}_v \in FN(\bar{R}_u)$  or both  $R_j \in BN(R_i)$  and  $\bar{R}_v \in BN(\bar{R}_u)$ .

Definition 4 states that the mapping of  $R_i$  to  $\bar{R}_j$  favors all possible mappings of forward (backward) neighbors of  $R_i$  to those of  $\bar{R}_j$ . For instance, if  $FN(R_i) = 2$ ,  $FN(\bar{R}_j) = 2$ ,  $BN(R_i) = 1$  and  $BN(\bar{R}_j) = 2$ , then the mapping  $(R_i, \bar{R}_j)$  supports  $2 \times 2 + 1 \times 2 = 6$  mappings. We distribute the support of  $(R_i, \bar{R}_j)$  equally to these six mappings. There can be cases when one mapping does not provide support to any others. In such cases, we simply distribute its support equally to all possible mappings ( $nM + mN - nm$ ). Conceptually, we consider the support of each mapping  $(R_i, \bar{R}_j)$  on the other mappings as a matrix. We call it *the support matrix* ( $S$ ) since it stores the topological support between different mappings. Notice that we are setting the entries of  $S$  in a way that for each mapping the sum of the relative weights of its support is 1. In other words, the sum of all the entries in each column of  $S$  is one. This ensures the stability and convergence of our algorithm as we explain in Section 2.4. Interested reader can find detailed description of the support matrix in a previous work of ours [8].

Trivial but costly way of creating  $S$  matrix is to check each mapping against all the others to calculate the support values. However, such an exhaustive strategy will require computing a huge matrix  $S$  of size  $(nM + mN - nm) \times (nM + mN - nm)$ . Since the creation of  $S$  will incur prohibitive computational costs, we do not construct this matrix literally. Instead, for each mapping  $(R_i, \bar{R}_j)$ , we take the sets  $FN(R_i)$ ,  $FN(\bar{R}_j)$  and  $BN(R_i)$ ,  $BN(\bar{R}_j)$  to generate only the pairs supported by  $(R_i, \bar{R}_j)$ . In other words, we use the sparse matrix form of the support matrix  $S$ .

## 2.4 Aligning Two Pathways

Both the homological similarities of subnetworks and their topological organization provide us significant information for the alignment of metabolic pathways. A good alignment algorithm needs to combine these two factors in an efficient and accurate way. Here, we describe how we achieve this combination in SubMAP by using an iterative technique called *power method*.

Let  $k$  be a given parameter and  $\mathcal{P}$ ,  $\bar{\mathcal{P}}$  be two pathways with connected subnetwork sets  $\mathcal{R}_k = \{R_1, R_2, \dots, R_N\}$  and  $\bar{\mathcal{R}}_k = \{\bar{R}_1, \bar{R}_2, \dots, \bar{R}_M\}$  respectively. We represent the homological similarity of all subnetwork pairs by the column vector  $\vec{H}$  of size  $nM + mN - nm$ , where  $n, m$  are the number of reactions of  $\mathcal{P}$ ,  $\bar{\mathcal{P}}$  respectively and  $N = |\mathcal{R}_k|$ ,  $M = |\bar{\mathcal{R}}_k|$ . Each entry of  $\vec{H}$  denotes the homological similarity between two subnetworks one from each pathway which corresponds to a mapping.



Let  $S$  be the  $(nM + mN - nm) \times (nM + mN - nm)$  support matrix as described in Section 2.3. Given a parameter  $\alpha \in [0, 1]$  to adjust the relative weights of homology and topology, we combine homology and topology through power method iterations as follows:

$$\overrightarrow{H^{k+1}} = \alpha S \overrightarrow{H^k} + (1 - \alpha) \overrightarrow{H^0} \quad (1)$$

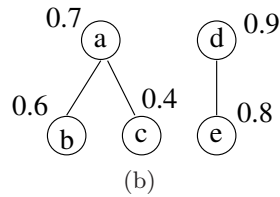
In this equation,  $\overrightarrow{H^0} = \overrightarrow{H}$ . We iterate this equation till  $\overrightarrow{H^{k+1}} = \overrightarrow{H^k}$  (i.e., it converges). The resulting vector  $\overrightarrow{H^p}$  is the principal eigenvector of the matrix  $\alpha S + (1 - \alpha) \overrightarrow{H^0} e$  where  $e$  is a row vector of size  $nM + mN - nm$  with all entries equal to 1. This system converges to a unique principal eigenvector when the matrices  $S$  and  $\overrightarrow{H^0} e$  are both column stochastic. We assure this as each column of  $S$  as well as  $\overrightarrow{H^0}$  itself adds up to one. Each entry of  $\overrightarrow{H^p}$  gives us a combination of homological and topological similarities for the corresponding mapping. We use  $\alpha = 0.6$  in this paper since in our previous work we observed that this value provides a good combination of the two similarities [7][8].

Recall that our aim is to find the relation  $\varphi$  that *maximizes* the summation of the similarity scores defined by  $\overrightarrow{H^p}$  while preserving the consistency between mappings. Using  $\overrightarrow{H^p}$  and the definition of conflict between two mappings (Definition 2), we create a vertex weighted undirected graph  $G_c = (V_c, E_c, w)$ , which we name as *the conflict graph* as follows. Each mapping  $(R_i, \bar{R}_j) \in \varphi$  corresponds to a vertex in  $V_c$ . We set the weight of each vertex  $a = (R_i, \bar{R}_j)$  (i.e.,  $w(a)$ ) to the similarity between  $R_i$  and  $\bar{R}_j$  as computed in  $\overrightarrow{H^p}$ . Since the number of possible one-to-many mappings is  $nM + mN - nm$ , the conflict graph has  $nM + mN - nm$  vertices (i.e.,  $|V_c| = nM + mN - nm$ ). We draw an undirected edge between two vertices  $a = (R_i, \bar{R}_j)$  and  $b = (R_u, \bar{R}_v)$  if  $(R_i \cap R_u) \cup (\bar{R}_j \cap \bar{R}_v) \neq \emptyset$  (i.e.,  $a$  and  $b$  conflict). For instance, in Figure 2 there is an edge between  $a$  and  $b$  representing that they conflict since reaction  $r_1$  is common to both  $a$  and  $b$ .

Extracting the subset of vertices that do not conflict (i.e., no edges) and maximize the sum of the similarity score from the conflict graph is equivalent to finding its *the maximum weight independent set (MWIS)*. MWIS problem can be reduced to our problem of finding the consistent alignment by simply mapping

	Subnetwork 1 ( $\mathcal{P}$ )	Subnetwork ( $\bar{\mathcal{P}}$ )	$\overrightarrow{H^p}$
a:	$R_1 = \{r_1, r_2\}$	$\bar{R}_1 = \{\bar{r}_1\}$	0.7
b:	$R_2 = \{r_1\}$	$\bar{R}_2 = \{\bar{r}_2\}$	0.6
c:	$R_3 = \{r_3\}$	$\bar{R}_3 = \{\bar{r}_1\}$	0.4
d:	$R_4 = \{r_4\}$	$\bar{R}_4 = \{\bar{r}_3, \bar{r}_4, \bar{r}_5\}$	0.9
e:	$R_5 = \{r_4, r_5\}$	$\bar{R}_5 = \{\bar{r}_5\}$	0.8

(a)



**Fig. 2.** (a) Each row corresponds to a possible mapping between subnetworks from two hypothetical metabolic pathways. The first column is the unique label for each mapping. Second and third columns are the reactions in the two subnetworks that can map. Last column is the similarity between the two subnetworks. (b) The conflict graph  $G_c$  for the mappings in (a).



each vertex to a mapping and each undirected edge to a conflict between two mappings. The MWIS problem is NP-hard [26] and there is no constant factor approximation to the optimal solution unless  $P = NP$  [27]. Therefore, we need a heuristic algorithm to find the MWIS of  $G_c$  and hence our alignment.

We adopt the greedy heuristic described by Sakai *et al.* [28]. Let  $N(v)$  denote the set of vertices that are connected to  $v$ . At each iteration of this algorithm, we pick the vertex  $v$  that maximizes  $f(v) = \sum_{u_i \in N(v)} \frac{w(v)}{w(u_i)}$ . This strategy implies that a vertex is more likely to be picked if the mapping it represents has large similarity score and conflicts with small number of other mappings with small similarity scores. After picking a vertex  $v$ , we put  $v$  into the result set and remove  $v$  and all the vertices connected to it ( $v \cup N(v)$ ). We also remove all the edges incident to at least one of the vertices in ( $v \cup N(v)$ ). When there are no more vertices to remove from  $G_c$ , the result set contains the vertices of a maximal weight independent set. For our alignment problem, this vertex set corresponds to a set of non-conflicting subnetwork mappings. As an example, in Figure 2,  $d$  is the first vertex to be picked. Then, we remove  $d$  and  $e \in N(d)$  from the graph and put  $d$  in the result set. Next, we pick the vertex  $b$  as  $f(b) = \frac{0.6}{0.7} > f(a) = \frac{0.7}{0.6+0.4} > f(c) = \frac{0.4}{0.7}$ . We remove  $b$  and  $a \in N(b)$  and put  $b$  in the result set. Finally, only  $c$  is left and taking it into our result set, we have our consistent alignment as the mappings  $b = (r_1, \bar{r}_2)$ ,  $c = (r_3, \bar{r}_1)$  and  $d = (r_4, \{\bar{r}_3, \bar{r}_4, \bar{r}_5\})$ .

### 3 Experimental Results

In this section, we experimentally evaluate the performance of SubMAP.

**Dataset:** We use the metabolic pathways of 20 organisms taken from the KEGG database. Our dataset contains 1,842 pathways in total. The average number of reactions per pathway is 21 and the largest pathway has 72 reactions.

#### 3.1 Alternative Subnetworks

Different organisms can perform the same function through different subnetworks. We name such altered parts that have similar functions as *alternative subnetworks*. An accurate alignment should reveal alternative subnetworks in different pathways. In our first experiment we evaluate whether SubMAP can find them in real metabolic pathways. We align the pathway pairs which are known to contain functionally similar parts with different reaction sets and topologies. Table 1 presents a subset of mappings that are found by our algorithm.

The first row of Table 1 corresponds to alternative subnetworks in Figure 1. The reaction R07613 represents the bottom path in Figure 1 that plants and Chlamydia use to produce LL-2,6- Diaminopimelate from 2,3,4,5- Tetrahydrodipicolinate. This path is discovered and reported as a shortcut on the L-Lysine synthesis path for plants and Chlamydia which is not present in humans [13,23]. Watanabe *et al.* [13] also suggest that since humans lack this path and hence the catalyzer of the reaction R07613, namely LL-DAP aminotransferase (EC:2.6.1.83), this is an

**Table 1.** Alternative subnetworks that produce same or similar output compounds from the same or similar input compounds in different organisms. <sup>1</sup> Main input compound utilized by the given set of reactions. <sup>2</sup> Main output compound produced by the given set of reactions. <sup>3</sup> Reactions mappings that corresponds to alternative paths. Reactions are represented by their KEGG identifiers.

Pathway	Organisms	Input Comp. <sup>1</sup>	Output Comp. <sup>2</sup>	Reaction Mappings <sup>3</sup>
Lysine biosynthesis	<i>A.thaliana</i> <i>E.coli</i>	2,3,4,5-Tetrahydrodipico.	LL-2,6-Di-aminopimelate	R07613 ⇔ R02734 + R04365 + R04475
Lysine biosynthesis	<i>A.thaliana</i> <i>E.coli</i>	L-Saccharopine meso-2,6-Di.	L-Lysine	R00451 + R00715 + R00716 ⇔ R00451
Pyruvate metabolism	<i>E.coli</i> <i>H.sapiens</i>	Pyruvate	Oxaloacetate	R00199 + R00345 ⇔ R00344
Pyruvate metabolism	<i>E.coli</i> <i>H.sapiens</i>	Oxaloacetate	Phosphoenolpyruvate	R00341 ⇔ R00431 + R00726
Pyruvate metabolism	<i>T.acidophilum</i> <i>A.tumefaciens</i>	Pyruvate	Acetyl-CoA	R01196 ⇔ R00472 + R00216 + R01257
Glycine, serine, threonine metabolism	<i>H.sapiens</i> <i>R.norvegicus</i>	Glycine	Serine L-Threonine	R00945 ⇔ R00751 + R00945 + R06171
Fructose and mannose metabolism	<i>E.coli</i> <i>H.sapiens</i>	L-Fucose	L-Fucose 1-p L-Fuculose 1-p	R03163 + R03241 ⇔ R03161
Citrate cycle	<i>S.aureus N315</i> <i>S.aureus COL</i>	Isocitrate	2-Oxoglutarate	R00268 + R01899 ⇔ R00709
Citrate cycle	<i>H.sapiens</i> <i>A.tumefaciens</i>	Succinate	Succinyl-CoA	R00432 + R00727 ⇔ R00405
Citrate cycle	<i>H.sapiens</i> <i>A.tumefaciens</i>	Isocitrate Citrate	2-Oxoglutarate Oxaloacetate	R00709 ⇔ R00362

attractive target for the development of new drugs (antibiotics and herbicides). When we align the Lysine biosynthesis pathways of *H.sapiens* and *A.thaliana* (a plant), our algorithm mapped the reaction R07613 of *A.thaliana* to the three reactions that *H.sapiens* has to use to transform 2,3,4,5- Tetrahydrodipicolinate to LL-2,6- Diaminopimelate (R02734, R04365, R04475). In other words, SubMAP successfully identified the alternative subnetworks of different size (1 for *A.thaliana* and 3 for *H.sapiens*) that perform the same function.

Another interesting example is the second row that is extracted from the same alignment described above. In this case, the three reactions that can produce L-Lysine for *A.thaliana* are aligned to the only reaction that produces L-Lysine for *H.sapiens*. R00451 is common to both organisms and it utilizes meso-2,6-Diaminopimelate to produce L-Lysine. The reactions R00715 and R00716 take place and produce L-Lysine in *A.thaliana* in the presence of L-Saccharopine [29].

For the alignment of Pyruvate metabolisms of *E.coli* and *H.sapiens*, the third and fourth rows show two mappings that are found by SubMAP. The first one maps the two step process in *E.coli* that first converts Pyruvate to Orthophosphate (R00199) and then Orthophosphate to Oxaloacetate (R00345) to the single reaction that directly produces Oxaloacetate from Pyruvate (R00344) in *H.sapiens*. The second one shows another mapping in which a single reaction of *E.coli* is replaced by two reactions of *H.sapiens*. The first two rows for Citrate cycle also report similar mappings for other organism pairs.

Note that all the above examples are one-to-many reaction mappings and hence a merit of the new algorithm we propose here. Our algorithm SubMAP also reports one-to-one mappings. The last row of Table 1 is an example in which one reaction of an organism is replaced by exactly one reaction of another organism. Aligning Citrate cycles of *H.sapiens* and *A.tumefaciens* reveals that

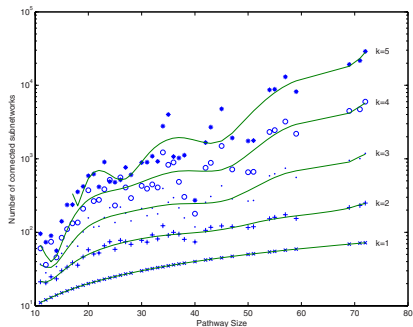
even though both the input and output compounds of two reactions R00709 and R00362 are different SubMAP maps these reactions. Also, if we look at the EC numbers of the enzymes catalyzing these reactions (1.1.1.41 and 4.1.3.6) their similarity is zero (see Information content enzyme similarity [8]). If we were to consider only the homological similarities, these two reactions could not have been mapped to each other. However, both these reactions are the neighbors of two other reactions R01325 and R01900 that are present in both organisms. The mappings of R01325 to R01325 and R01900 to R01900 support the mapping of their neighbors R00709 to R00362. Therefore, by incorporating the topological similarity our algorithm is able to find meaningful mappings with similar topologies and distinct homologies. An algorithm not considering pathway topologies would fail to identify such mappings.

*These results suggest: (i) By allowing one-to-many mappings, our method identifies functionally similar subnetworks even if they have different number of reactions. (ii) The incorporation of topological similarity makes it possible to find mappings that can be missed by only considering homological similarity.*

### 3.2 Number of Connected Subnetworks

Given the parameter  $k$ , our algorithm enumerates all connected reaction subnetworks of size at most  $k$  for each query pathway. One question that we need to answer is: How many such subnetworks exist? Figure 3 plots this number for all the pathways in our dataset. When  $k = 1$ , the figure shows the number of reactions in each pathway. For  $k > 1$  the results demonstrate that the number of subnetworks increase exponentially with  $k$ . However, the increase is significantly lower than the theoretical worst case  $\sum_{i=1}^k \binom{n}{i}$  (i.e.,  $n$  choose  $i$ ). For instance, the largest number of subnetworks we obtained for  $n = 72$  and  $k = 5$  is around 750 times less than the theoretical worst case.

The figure also suggests that the number of subnetworks increase linearly with the size of the pathway. This is mainly because the average number of edges (i.e., neighbors) of a node (i.e., subnetwork) remains roughly same as the size of the network increases. As a result, we conclude that for  $k \leq 4$ , we can enumerate and store all the subnetworks in our dataset. The number of subnetworks for  $k = 5$  is still small enough to handle. However, in practice it is unlikely for a single reaction to replace a subnetwork with such a large number of reactions. We expect that  $k \leq 4$  would be sufficient to find most of the alternative subnetworks. Hence, we use  $k \leq 4$  in our experiments.



**Fig. 3.** The number subnetworks with at most  $k$  nodes for pathways of different sizes

### 3.3 One-to-Many Mappings within and across Major Clades

In Section 3.1, we demonstrated that our algorithm can find alternative subnetworks on a number of examples. An obvious question that follows is: How frequent are such alternative subnetworks and what are their characteristics? In other words, is there really a need to allow one-to-many mappings in alignment. In this experiment we aim to answer these questions.

We conduct an experiment as follows.

We first pick 9 different organisms 3 from each major phylogenetic clade. These organisms are *T.acidophilum*, *Halobacterium sp.*, *M.thermoautotrophicum* from Archaea; *H.sapiens*, *R.norvegicus*, *M.musculus* from Eukaryota; and *E.coli*, *P.aeruginosa*, *A.tumefaciens* from Bacteria. We then extract 10 common pathways for these 9 organisms from KEGG. For each of these common pathways, we choose all possible pairs of the 9 organisms ( $\binom{9}{2} = 36$ ) and align that specific pathway for all organism pairs. In these alignments we exclude the self alignments and the

alignment with parameter  $k = 1$  since those will definitely incur a bias favoring the number of one-to-one alignments. We computed all possible alignments ( $10 \times 36 = 360$ ) for  $k = 2, 3$  and 4 ( $360 \times 3 = 1,080$  alignments in total). Finally, we calculated the number of four possible types of subnetwork mappings which are 1-to-1, 1-to-2, 1-to-3 and 1-to-4. We hypothesize that the metabolisms of the organisms within a clade will tend to perform the same function through the same (or similar) sized sets of reactions while those across different clades will perform from alternative subnetworks of varying sizes.

Table 2 summarizes the results of this experiment. The percentages of each mapping type between two clades is shown as a row in this table. The first three rows corresponds to alignments within a clade and the last three represents alignments across two different clades. An important outcome of these results is that there are considerably large number of one-to-many mappings between organisms of different clades. In the extreme case (last row), nearly half of the mappings are one-to-many. The results also support our hypothesis that one-to-one mappings is more frequent for alignments within the clades compared to across clades due to high similarity between the organisms of the same clade. For instance, for both the first and last row one side of the query set is the Eukaryota. However, going from first row to last, we see around 40% decrease in the number of one-to-one mappings and 250%, 850% and 450% increase in the number of 1-to-2, 1-to-3 and 1-to-4 mappings respectively. Considering Archaea are single-celled microorganisms (e.g., Halobacteria) and Eukaryota are complex organisms with cell membranes (e.g., animals and plants), these jumps in the number of one-to-many mappings suggest that the individual reactions in Archaea are replaced by a number of reactions in Eukaryota. These results

**Table 2.** Percentages of 1-to-1, 1-to-2, 1-to-3 and 1-to-4 mappings in between and across three major clades (**A**: Archaea, **E**: Eukaryota, **B**: Bacteria)

	1-to-1	1-to-2	1-to-3	1-to-4
<b>E-E</b>	89.6	8.8	1.1	0.5
<b>B-B</b>	80.1	16.0	3.1	0.8
<b>A-A</b>	78.3	15.7	4.7	1.3
<b>B-E</b>	69.1	23.1	6.3	1.5
<b>A-B</b>	60.5	28.3	8.5	2.7
<b>A-E</b>	55.8	31.0	10.4	2.8

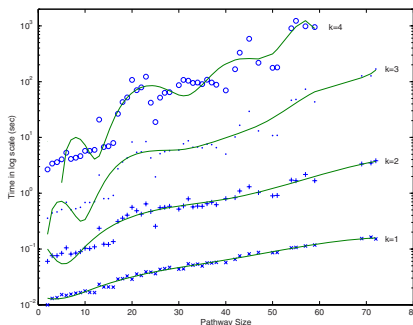
have two major implications. (i) *One-to-many mappings are frequent in nature. To obtain biologically meaningful alignments we need to allow such mappings.* (ii) *The characteristics of the alternative subnetworks can help in inferring the phylogenic relationship among different organisms.*

### 3.4 Evaluation of Running Time and Memory Utilization

SubMAP allows one to many mappings to find biologically relevant alignments. This however comes at the expense of increased computational cost. Theoretically, this increase can be exponential in  $k$  in the worst case. The worst case happens when the pathway is highly connected. Metabolic pathways however are sparse and their connectivity follows power law distribution [30]. In order to understand the capabilities and limitations of our method we examine its performance on real datasets in terms of its running time and memory usage.

We evaluate the performance of our method for querying a database of pathways as follows. We create a query set by selecting 50 pathways of varying sizes from our dataset described at the beginning of this section. We then select another 50 pathways of different sizes to use as our database set for this experiment. We pick the latter 50 pathways such that the average reactions per pathway is 21.4, which is very close to that of the entire database. We then align each query pathway with all the database pathways one by one for different values of  $k$ . We measure the average running time and the average memory usage for each query pathway and  $k$  value combination. Note that we do not present any performance comparison with an existing method as the existing methods do not allow one-to-many mappings. However, our results for  $k = 1$ , shows the performance of our algorithm when we restrict it to one-to-one mappings.

Figure 4 shows the average running time of SubMAP for query pathways with increasing number of reactions. When  $k = 1$  (i.e., only one-to-one mappings as in existing methods), it runs in less than 0.2 seconds even for the largest query pathway in our query set. As  $k$  increases, the running time increases as well. This is because the number of subnetworks and the average numbers of forward and backward neighbors of subnetworks increase with  $k$ . However, we observe that our method can perform alignments in practical time even when  $k = 4$ . It aligns pathways with around 50 reactions in less than one minute and 20 minutes for  $k = 3$  and 4 respectively. It runs in less than 15 minutes for the largest query pathway (72 reactions) in our query set for  $k = 3$ .



**Fig. 4.** The average running time of SubMAP when a query pathway is aligned with all the pathways in a pathway database. The selected pathway database contains 50 pathways. X-axis is the number of the reactions of the query pathways.

We also measure the actual memory usage of our algorithm for real pathways of varying sizes and  $k$  values (Figure omitted). For  $k = 1$  or  $2$ , the memory usage is negligible (1 MB or less) for all pathways. Although the memory usage increases with  $k$ , it remains feasible even for query pathways with around 50 reactions for  $k = 4$ . Our algorithm uses less than 300 MB for the largest query when  $k = 3$ . For two query pathways both with around 50 reactions and  $k = 4$ , the memory requirement is around 600 MB. *Thus, our algorithm can run on a standard computer for aligning real-sized metabolic pathways.*

## 4 Conclusion

In this paper, we considered the problem of aligning two metabolic pathways. The distinguishing feature of our work from the literature is that we allow mapping one molecule of one pathway to a set of molecules of the other. To address this problem, given two metabolic pathways  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  and an upper bound  $k$  on the size of the connected subnetworks, we developed the SubMAP algorithm that can find the consistent mapping of the subnetworks of  $\mathcal{P}$  and  $\bar{\mathcal{P}}$  with the maximum similarity. We transformed the alignment problem to an eigenvalue problem. The solution to this eigenvalue problem produced a good mixture of homological and topological similarities of the subnetworks. Using these similarity values, we constructed a vertex weighted graph that connects conflicting mappings with an edge. Then, our alignment problem transformed into finding the maximum weight independent subset of this graph. We employed a heuristic method that is used to solve maximum weight independent set problem. The result of this method provided us an alignment that has no conflicting pair of mappings (i.e., consistent). Our experiments on real datasets suggested that our method can identify biologically relevant alignments of alternative subnetworks that are missed by traditional methods. Furthermore, even though SubMAP does not restrict the topologies of query pathways, it is still scalable for real size metabolic pathways when the reaction subsets of size at most four are considered.

## References

1. Edwards, J.S., Palsson, B.O.: Robustness analysis of the Escherichia coli metabolic network. *Biotechnology Progress* 16, 927–939 (2000)
2. Ay, F., Xu, F., Kahveci, T.: Scalable Steady State Analysis of Boolean Biological Regulatory Networks. *PLoS ONE* 4(12), e7992 (2009)
3. Schuster, S., Pfeiffer, T., Koch, I., Moldenhauer, F., Dandekar, T.: Exploring the Pathway Structure of Metabolism: Decomposition into Subnetworks and Application to Mycoplasma pneumoniae. *Bioinformatics* 18, 351–361 (2002)
4. Koyuturk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. In: *ISMB*, pp. 200–207 (2004)
5. Qian, X., Yoon, B.: Effective Identification of Conserved Pathways in Biological Networks Using Hidden Markov Models. *PLoS ONE* 4(12), e8070 (2009)
6. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. *Bioinformatics* 21(16), 3401–3408 (2005)

7. Ay, F., Kahveci, T., de Crecy-Lagard, V.: Consistent alignment of metabolic pathways without abstraction. In: Computational Systems Bioinformatics Conference (CSB), vol. 7, pp. 237–248 (2008)
8. Ay, F., Kahveci, T., de Crecy-Lagard, V.: A fast and accurate algorithm for comparative analysis of metabolic pathways. *Journal of Bioinformatics and Computational Biology (JBCB)* 7(3), 389–428 (2009)
9. Tohsato, Y., Nishimura, Y.: Metabolic Pathway Alignment Based on Similarity of Chemical Structures. *Information and Media Technologies* 3, 191–200 (2008)
10. Tohsato, Y., Matsuda, H., Hashimoto, A.: A Multiple Alignment Algorithm for Metabolic Pathway Analysis Using Enzyme Hierarchy. In: ISMB, pp. 376–383 (2000)
11. Cheng, Q., Harrison, R., Zelikovsky, A.: MetNetAligner: a web service tool for metabolic network alignments. *Bioinformatics* 25(15), 1989–1990 (2009)
12. Sridhar, P., Kahveci, T., Ranka, S.: An iterative algorithm for metabolic network-based drug target identification. In: Pacific Symposium on Biocomputing (PSB), vol. 12, pp. 88–99 (2007)
13. Watanabe, N., Cherney, M.M., van Belkum, M.J., Marcus, S.L., Flegel, M.D., Clay, M.D., Deyholos, M.K., Vederas, J.C., James, M.: Crystal structure of LL-diaminopimelate aminotransferase from *Arabidopsis thaliana*: a recently discovered enzyme in the biosynthesis of L-lysine by plants and Chlamydia. *Journal of Molecular Biology* 371(3), 685–702 (2007)
14. Francke, C., Siezen, R.J., Teusink, B.: Reconstructing the metabolic network of a bacterium from its genome. *Trends in Microbiology* 13(11), 550–558 (2005)
15. Clemente, J.C., Satou, K., Valiente, G.: Reconstruction of Phylogenetic Relationships from Metabolic Pathways Based on the Enzyme Hierarchy and the Gene Ontology. *Genome Informatics* 16(2), 45–55 (2005)
16. Heymans, M., Singh, A.: Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics* 19, 138–146 (2003)
17. Ogata, H., Fujibuchi, W., Goto, S., Kanehisa, M.: A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research* 28, 4021–4028 (2000)
18. Green, M.L., Karp, P.: A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC Bioinformatics* 5, 76 (2004)
19. Damaschke, P.: Graph-Theoretic Concepts in Computer Science. In: Möhring, R.H. (ed.) WG 1990. LNCS, vol. 484, pp. 72–78. Springer, Heidelberg (1991)
20. Webb, E.C.: Enzyme nomenclature 1992. Academic Press, London (1992)
21. Singh, R., Xu, J., Berger, B.: Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 16–31. Springer, Heidelberg (2007)
22. Deutscher, D., Meilijson, I., Schuster, S., Ruppin, E.: Can single knockouts accurately single out gene functions? *BMC Systems Biology* 2, 50 (2008)
23. McCoy, A.J., Adams, N.E., Hudson, A.O., Gilvarg, C., Leustek, T., Maurelli, A.T.: L,L-diaminopimelate aminotransferase, a trans-kingdom enzyme shared by Chlamydia and plants for synthesis of diaminopimelate/lysine. *PNAS* 103(47), 17909–17914 (2006)
24. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 27(1), 29–34 (1999)



25. Hattori, M., Okuno, Y., Goto, S., Kanehisa, M.: Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society (JACS)* 125(39), 11853–11865 (2003)
26. LovGasz, L.: Stable set and polynomials. *Discrete Mathematics* 124, 137–153 (1994)
27. Austrin, P., Khot, S., Safra, M.: Inapproximability of Vertex Cover and Independent Set in Bounded Degree Graphs. In: *IEEE Conference on Computational Complexity*, pp. 74–80 (2009)
28. Sakai, S., Togasaki, M., Yamazaki, K.: A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Applied Mathematics* 126, 313–322 (2003)
29. Saunders, P.P., Broquist, H.: Saccharopine, an intermediate of amino adipic acid pathway of lysine biosynthesis. *Journal of Biological Chemistry* 241, 3435–3440 (1966)
30. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabasi, A.: The large-scale organization of metabolic networks. *Nature* 407(6804), 651–654 (2000)



# Admixture Aberration Analysis: Application to Mapping in Admixed Population Using Pooled DNA

Sivan Bercovici and Dan Geiger

Computer Science Department  
Technion, Haifa 32000, Israel  
{sberco, dang}@cs.technion.ac.il

**Abstract.** Admixture mapping is a gene mapping approach used for the identification of genomic regions harboring disease susceptibility genes in the case of recently admixed populations such as African Americans. We present a novel method for admixture mapping, called admixture aberration analysis (AAA), that uses a DNA pool of affected admixed individuals. We demonstrate through simulations that AAA is a powerful and economical mapping method under a range of scenarios, capturing complex human diseases such as hypertension and end stage kidney disease. The method has a low false-positive rate and is robust to deviation from model assumptions. Finally, we apply AAA on 600 prostate cancer-affected African Americans, replicating a known risk locus. Simulation results indicate that the method can yield over 96% reduction in genotyping. Our method is implemented as a Java program called *AAAmap* and is freely available.

## 1 Introduction

Many complex disease studies are currently being conducted using population-based genetic association [1]. The premise of this method is that affected individuals carry a common variant of a disease susceptible gene which is in linkage disequilibrium with sampled markers. Hence, the susceptibility locus can be detected via the indirect association between the sampled markers and the disease status. In order to guarantee a sufficiently high power in association studies, thousands of cases and controls are sampled using dense marker panels.

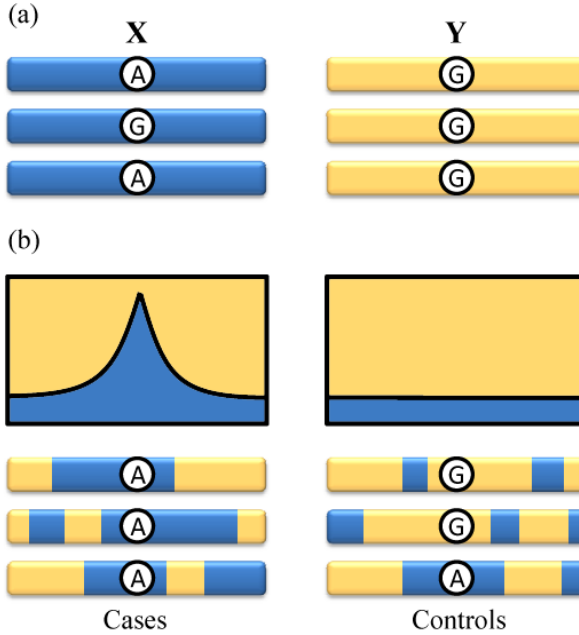
Admixture mapping, also known as Mapping by Admixture Linkage Disequilibrium (MALD), offers a more economical alternative to association studies in certain circumstances without sacrificing the statistical power [2]. MALD is a gene mapping approach used for the identification of genomic regions harboring disease susceptibility genes in the case of recently admixed populations, i.e. populations that are an admixture of several ancestral populations. African Americans are an example of an admixed population, having both European and African ancestries. The method is applicable when the prevalence of a disease is significantly different between the ancestral populations from which the

admixed population was formed. When such a disease is studied, admixed individuals carrying the hereditary disease are expected to show an elevated genomic contribution from the ancestral population that has the higher prevalence of the disease around the disease gene loci. A MALD study is comprised of three main steps. First, a panel of ancestry informative markers (AIM) that differentiate well between ancestral populations is designed. Next, either cases or both cases and controls are individually genotyped using the AIM panel, and the mosaic of ancestries of each individual is inferred. Finally, the inferred ancestral profiles are scanned in search for an aberration towards the ancestral population with the higher risk, as expected to appear near the disease locus.

The MALD method successfully discovered multiple risk alleles for prostate cancer [34], a disease with a higher incident rate in Africans compared to Europeans, and a candidate locus for end-stage kidney disease in African Americans [5]. Diseases of similar characteristics include stroke, hypertension and multiple-sclerosis; a more comprehensive list of diseases suitable for admixture mapping appears in the method's review by Smith and O'Brien [2]. In all of these cases, the statistical efficiency of MALD stems from the fact that only a few thousands of ancestry informative markers are required in order to accurately infer the ancestry of the admixed individuals [6,7]. Moreover, only a few hundreds of cases are required for the identification of the ancestral aberration around the disease locus [8].

In this paper we present a novel approach for admixture mapping that considerably reduces the genotyping cost of disease studies by applying admixture aberration analysis (AAA) on pooled DNA of affected admixed individuals. Our analysis detects divergence of allele distribution in a pool of samples near a disease locus without the intermediate step of ancestry inference per individual. The inherent aberration in admixture around the disease locus shifts the sampled allele frequencies towards the distribution of the alleles in the ancestry with the higher risk. It is the examination of this shift, evaluated through the estimation of allele frequencies in the pooled sample, that provides the means for our pooled mapping method. Figure 1 illustrates this idea.

Current MALD studies mainly differ in the informative panel of choice and the method used for ancestry inference. Patterson *et al.* [9] presented a method that employs a hidden Markov model (HMM) for the estimation of ancestry along the genome. The HMM was integrated into a Markov chain Monte Carlo (MCMC) method to account for the uncertainties in model parameters. Tang *et al.* [10] extended previous methods by modeling linkage-disequilibrium in the ancestral populations using a Markov Hidden Markov model (MHMM), namely, dependency between adjacent markers evident in the ancestral populations was modeled. An inference framework developed in [11] enables the incorporation of more complex probability models that account for linkage disequilibrium in the ancestral populations. An earlier work by Chakraborty and Weiss [12] suggested mapping by directly assessing divergence from admixture linkage-disequilibrium, as expected near disease loci.



**Fig. 1.** An illustration of admixture aberration. (a) Two distinct ancestral populations,  $X$  and  $Y$ , expressing a different distribution of alleles at a particular location. The greater the distance between allele distributions, the more informative the marker is regarding ancestry. (b) a sample of admixed individuals, descendants of the ancestral populations  $X$  and  $Y$ . In case of a disease with higher prevalence in population  $X$ , the affected sample will exhibit a higher contribution from population  $X$  near the disease locus, as indicated by the graph on the left. Hence, in the affected individuals, the distributions of alleles near the disease locus bears a higher resemblance to that of population  $X$ . The healthy admixed individuals show a contribution of populations  $X$  and  $Y$  that corresponds to the admixture process.

DNA Pooling has been suggested as a practical way to reduce the cost of large-scale association studies [13]. Rather than analyzing thousands of cases and controls that were sampled separately, association analysis was first applied on pooled cases and pooled controls in the work of Arnheim *et al.* [14]. Steer *et al.* [15] have recently demonstrated the feasibility of pooled association studies using high resolution microarrays for rheumatoid arthritis. Zeng and Lin [16] examined the analysis of pooled DNA, extending the single-marker association methods to haplotype association using a likelihood-based approach. Kirov *et al.* [17] investigated the accuracy by which the allele frequency difference between pools can be estimated. This work was extended by Wilkening *et al.* [18] for higher resolution SNP microarrays of 250K. Pooling was also used in QTL studies. For example, Darvasi and Soller [19] presented a statistical test of marker-QTL linkage based on selective pools of individuals with extreme quantitative trait values.

The main contribution of this paper is the introduction of pooling to admixture mapping, and the demonstration of its power to the mapping of disease susceptibility loci. Pooling is a far more effective tool for admixture mapping in comparison to association studies. In the case of a recently formed admixed population, the linkage-disequilibrium patterns generated by the admixture process stretch over regions of several centimorgans, resulting in a wider effect which is easier to detect. In addition, using ancestry informative markers improves the ability to locate deviations of LD and marker distribution from those expected by the admixture process alone. The efficiency of our pooled AAA method has been established through simulation and via analysis of diseases that are currently being studied using the non pooled MALD approach. Specifically, we first develop the aberration analysis method based on a window of markers while accounting for linkage disequilibrium in the ancestral populations. We then determine the method's power through simulations. We show, for example, that a power of over 70% is achieved in a simulated study of an African American population carrying a disease with ethnicity relative risk of 1.3, comparable with end-stage kidney disease, using 7 pools of 200 individuals with 4 repetitions. The results in this case indicate a more than 25-fold decrease in genotyping versus a non-pooled MALD method. We also demonstrate the strength of our pooled method on a sample of African American cases of prostate cancer, replacing 600 independently measured individuals with a single simulated pool. The result demonstrate that a significant signal (LOD 7.2) is obtained near the risk locus found by Amundadottir *et al.* [20] and Freedman *et al.* [3]. Finally, we discuss the robustness of our method to measurement errors and to deviation from model assumptions.

## 2 Material and Methods

### 2.1 Definitions and Model Assumptions

The genome of a recently admixed individual is a mosaic of long, single ancestry, chromosomal segments. We use the following definitions to describe these segments in admixed individuals. An *admixed chromosome* is a chromosome that originated from more than one ancestral population. A *Post Admixture Recombination* point (abbreviated PAR) is a recombination point in which either two chromosomes from different populations crossed, or two chromosomes crossed where at least one of the chromosomes is an admixed chromosome. A (*PAR*) *block* is a chromosomal segment limited by two consecutive PAR points, or by a chromosome edge and its closest PAR point. An immediate implication of these definitions is that every PAR block originated from a single ancestral population, designated as the ancestry of the block, for otherwise the block would have been further divided. In our model, we assume that the ancestry of PAR blocks are mutually independent. We further assume that given the ancestry of a PAR block, the markers within that PAR block are independent of the markers outside the PAR block and are determined strictly according to the distribution that corresponds to the ancestry of that PAR block [7]. The markers within a

PAR block are assumed to be dependent, accounting for the background linkage-disequilibrium in the ancestral populations.

Consider an admixed population that originated from two ancestral populations  $X$  and  $Y$ . Each ancestral population may have a different prevalence for a disease. A common way to characterize the disease risk attributed to the ancestral profile is by the ethnicity relative risk (ERR) which measures the increased risk due to an additional allele from population  $Y$ . Under a multiplicative disease model, ERR is defined as

$$r = \frac{\psi(XY)}{\psi(XX)} = \frac{\psi(YY)}{\psi(XY)} \quad (1)$$

where  $\psi(\cdot)$  is the probability of the disease given that the ancestry pair at the disease susceptibility locus is either  $XX$ ,  $XY$ , or  $YY$ .

When studying an admixed population with an hereditary disease characterized by an  $ERR \neq 1$ , the regions around the disease loci are expected to show an aberration towards the ancestry with the higher risk, shifting the distribution of nearby allele frequencies. Our method scans through the genome, computing for each examined location the ratio between the likelihood of the measured allele frequencies under the assumption of a close disease locus and the likelihood of the measured frequencies under the null assumption of no disease:

$$\Lambda_0 = \frac{P(S|\text{nearby disease locus})}{P(S|\text{no disease})} \quad (2)$$

where  $S$  are the observed allele frequencies. Since the computation of this likelihood becomes intractable as the number of samples and markers grow, we approximate these probabilities via the multivariate central limit theorem over a window of markers. This approximate measure, denoted  $\Lambda$ , is used in the reported results. In the remaining method section, we derive the distribution of alleles under the two hypothesis, and the  $\Lambda$  score. We first assume a window with a single marker and then extend the results to multi-marker windows.

## 2.2 Single Marker Analysis

We first compute the probability  $P(J|d)$  of a bi-allelic marker  $J \in \{0, 1\}$  of an individual, given the individual is affected (denoted by  $d$ ). This probability is given by

$$P(J|d) = P(J|\bar{r}, d) \cdot P(\bar{r}|d) + P(J|r, d) \cdot P(r|d) \quad (3)$$

where  $r$  indicates that at least one recombination has occurred between the disease locus and the location of allele  $J$  since the first admixture event, and  $\bar{r}$  is the complementary event.

The occurrence of post-admixture recombination points (PAR) can be modeled as a Poisson process with rate  $\lambda$  which is derived from the admixture dynamics. In the case of a hybrid-isolated admixture model [21],  $\lambda$  roughly corresponds

to the number of generations since the admixture began. Hence, under the assumption that the event of a recombination is independent of the disease status, the probability of at least one PAR point between location  $l_1$  and  $l_2$  is

$$P(r|d) = P(r) = 1 - e^{-\lambda \cdot |l_1 - l_2|} \quad (4)$$

To compute  $P(J|r, d)$  in Equation 3, we note that given  $r$ , namely that at least one PAR point occurred between sampled allele  $J$  and the disease locus, the distribution of the allele is determined solely by the ancestry at the location and the admixture coefficient  $P(Q)$ :

$$\begin{aligned} P(J|r, d) &= \sum_Q P(J|Q, r, d) \cdot P(Q|r, d) \\ &= \sum_Q P(J|Q) \cdot P(Q) \end{aligned} \quad (5)$$

where  $Q$  is the ancestry at the marker location.

To compute  $P(J|\bar{r}, d)$  in Equation 3, namely when assuming no PAR point exist between the disease locus and the sampled allele, the distribution of the allele is given by

$$P(J|\bar{r}, d) = \sum_{Q'} P(J|Q') \cdot P(Q'|d) \quad (6)$$

where  $Q'$  is the ancestry at the disease locus. The above equality relies on the assumption that given the ancestry of the chromosomal segment containing marker  $J$ , the affection status and the allele are independent, an assumption that is common in admixture mapping models 9. The probability  $P(Q'|d)$  of the ancestry of an affect individual at disease locus  $Q'$  is formalized in terms of the multiplicative disease model. Let  $Z' \in \{XX, XY, YY\}$  denote the ancestry pair at the disease locus. The probability of ancestry  $Q'$  given the disease can be written as

$$\begin{aligned} P(Q' = X|d) &= \sum_{Z'} P(Q' = X|Z', d) \cdot P(Z'|d) \\ &= P(Z' = XX|d) + \frac{P(Z' = XY|d)}{2} \end{aligned} \quad (7)$$

The probability  $P(Z' = XX|d)$  is computed from  $\psi(\cdot)$  as follows:

$$\begin{aligned} P(Z' = XX|d) &= \frac{P(D|Z' = XX) \cdot P(Z' = XX)}{\sum_{Z'} P(D|Z') \cdot P(Z')} \\ &= \frac{\psi(XX) \cdot p_X^2}{\psi(XX) \cdot p_X^2 + 2\psi(XY)p_X(1 - p_X) + \psi(YY)(1 - p_X)^2} \end{aligned}$$

where  $p_X$  is the a priori probability of ancestry  $X$  in an admixed individual. The probabilities  $P(Z' = XY|d)$  and  $P(Z' = YY|d)$  are derived in a similar fashion. This completes the derivation of all terms of Equation 3.

We continue by considering a set of independent marker observations  $J_1, J_2, \dots, J_n$  sampled from  $n$  affected admixed individuals. We need to compute the likelihood ratio  $\mathcal{L}$  of these observations, namely the probability of the observations under the hypothesis of a nearby disease susceptibility locus divided by the probability under the null hypothesis of no disease

$$\mathcal{L} = \frac{P(J_1, \dots, J_n | H_1)}{P(J_1, \dots, J_n | H_0)}$$

As we assume independent and identically distributed  $J_i$ , we conclude that

$$\binom{n}{|\{J_i | J_i = 1\}|} \cdot P(J_1, \dots, J_n) = P(S_n)$$

where  $S_n = \sum_i J_i$ . Hence, the likelihood ratio can be rewritten as follows

$$\mathcal{L} = \frac{P(J_1, \dots, J_n | H_1)}{P(J_1, \dots, J_n | H_0)} = \frac{P(S_n | H_1)}{P(S_n | H_0)}$$

We now explicate how to approximate the probabilities  $P(S_n | H_0)$  and  $P(S_n | H_1)$ .

According to the central limit theorem, the standardized sum of  $n$  observations converges to the standard normal distribution  $N(0, 1)$  as  $n$  grows

$$S_n^* = \frac{\sum J_i - n \cdot \mu}{\sigma \sqrt{n}} \rightarrow N(0, 1)$$

where  $\mu$  and  $\sigma$  are determined by the distribution of  $J$ . For the two hypotheses, we use the following means and variances:

$$\begin{aligned} \mu_0 &= P(J|r, d), \quad \sigma_0 = \sqrt{P(J|r, d) \cdot (1 - P(J|r, d))} \\ \mu_1 &= P(J|d), \quad \sigma_1 = \sqrt{P(J|d) \cdot (1 - P(J|d))} \end{aligned}$$

Note that  $P(J|d)$  is given by Equation [3](#), and that  $P(J|r, d)$  is given by Equation [5](#). The use of  $P(J|r, d)$  for the null hypothesis is justified because this case is equivalent to an infinitely distant disease locus. Each hypothesis yields a different distribution of the markers hence a different standardization, and in turn, a corresponding probability for the sum of observations. We denote the standardized sums of  $S_n$  according to hypotheses  $H_0$  and  $H_1$  by  $S_n^{H_0}$  and  $S_n^{H_1}$ , respectively. The likelihood ratio of the observations under the two hypothesis can now be approximated as follows

$$\mathcal{L} = \frac{P(J_1, \dots, J_n | H_1)}{P(J_1, \dots, J_n | H_0)} \rightarrow \frac{P(S_n^{H_1})}{P(S_n^{H_0})} = \Lambda \quad (8)$$

The  $\log_{10}$  of  $\Lambda$  is called the LOD score; high LOD scores are indicative of a nearby disease locus.

In the above derivation, we assumed that the  $n$  marker observations are independent even though each affected individual contributes two observations to the sample. The effect of this discrepancy weakens as the sample size increases.

As a final note consider the case of fully informative markers. Such markers have one allele with probability 1 in the first ancestral population and the other allele with probability 1 in the second ancestral population. When using fully informative markers and assuming no errors reading them, the ancestry at each marker location is known with certainty using a single marker readings. In this case, a non-pooled MALD locus statistic such as the one described in [9], reduces to the ratio between the probability of ancestry given a nearby disease locus and the a priori probability of ancestry (rather than a ratio between probabilities of marker data). This ratio exactly equals, in the limit of sufficiently large samples, to our  $\Lambda$  statistic under fully informative markers. Consequently, for sample sizes that one normally deals with in MALD studies ( $> 500$  samples), our AAA method retains the same statistical power as non-pooled MALD but at orders of magnitude less genotyping under this scenario. A comparison of the power of the two methods is further studied in Section 3 without assuming fully informative markers.

### 2.3 Multi-marker Analysis

We now extend our analysis from a single marker to the case of haplotypes where  $m$  bi-allelic markers are sampled. First, we derive the probability  $P(J|d)$  of an individual to carry haplotype  $J \in \{0, \dots, 2^m - 1\}$  given that the individual is affected (denoted by  $d$ ). This probability can be written via

$$P(J|d) = \sum_{\pi} P(J|\pi, d) \cdot P(\pi) \quad (9)$$

where  $\pi$  is a partition of the haplotype into PAR blocks. The probability of a partition  $p(\pi)$  is determined by the independent PAR points that either occurred or did not occur between sampled markers  $\prod_{i=1}^{m-1} P(R_i)$ , where the variable  $R_i \in \{0, 1\}$  denotes whether a PAR point occurred between markers  $i$  and  $i + 1$ , and the probability  $P(R_i = 1)$  is given by Equation 4.

To compute the remaining term  $p(J|\pi, d)$  in Equation 9, recall that our admixture model assumes that markers within a PAR block are independent of markers outside the PAR block given the ancestry of the block. Hence, given partition  $\pi$ , the probability of haplotype  $J$  is given by

$$\begin{aligned} P(J|\pi, d) &= \prod_b P(J_b|d) \\ &= \prod_b \sum_{Q_b} P(J_b|Q_b, d) \cdot P(Q_b|d) \end{aligned} \quad (10)$$

where  $b$  is a block in partition  $\pi$ ,  $J_b$  are the markers within block  $b$ , and  $Q_b$  is the ancestry of that block. The probability of a block's ancestry given an affected individual is determined by whether or not the disease locus is within the PAR block in question, hence is

$$P(Q_b|d) = \begin{cases} P(Q'|d) & l_d \in b \\ \pi_Q & \text{otherwise} \end{cases}$$



where  $l_d$  is the tentative disease locus,  $\pi_Q$  is the a prior probability of ancestry  $Q$ , and  $P(Q'|d)$  is given in Equation 7.

Our model assumes that given the ancestry of a block, the haplotype distribution is independent of the disease status. Hence, the term  $P(J_b|Q_b, d)$  in Equation 10 is equals the probability  $P(J_b|Q_b)$  which can be computed via samples taken from the ancestral populations. For example, European and West African individuals phased in the HapMap project [22] were used in Section 3 to construct the ancestral haplotype distribution  $P(J|Q)$  for the analysis of African American. This concludes the derivation of all the terms used in the computation of Equation 9.

Finally, we consider a set of independent haplotype observations  $J_1, J_2, \dots, J_n$  sampled from  $n$  affected admixed individuals. We compute the likelihood ratio of the pooled observations, dividing the probability under the hypothesis of a nearby disease susceptibility locus by the probability under the null hypothesis of no disease:

$$\mathcal{L} = \frac{P(S_n|H_1)}{P(S_n|H_0)}$$

where  $S_n$  is the sum of observations  $J_i$ .

We continue by explicating the computation of the probabilities  $P(S_n|H_0)$  and  $P(S_n|H_1)$ . According to the multivariate central limit theorem, under the assumption that the covariance matrix of  $J$  is positive-definite, the standardized sum of  $n$  observations converges towards the standard normal distribution  $N(0, \Sigma)$  as  $n$  grows

$$S_n^* = \frac{\sum J_i - n \cdot \mu}{\sqrt{n}} \rightarrow N(0, \Sigma)$$

where  $\mu$  and  $\Sigma$  are determined by the distribution of  $J$  assuming an affected admixed individual. For the two hypotheses, we use the following means and covariance matrices:

$$\begin{aligned} \mu_0 &= \sum_J J \cdot P(J|d, l_d = \infty) \\ \mu_1 &= \sum_J J \cdot P(J|d, l_d = l) \\ \Sigma_{i,j}^0 &= E\left((J^i - \bar{J}^i)(J^j - \bar{J}^j) \mid l_d = \infty\right) \\ \Sigma_{i,j}^1 &= E\left((J^i - \bar{J}^i)(J^j - \bar{J}^j) \mid l_d\right) \end{aligned}$$

where  $J^i$  indicates the  $i^{th}$  component of haplotype  $J$ . Under the alternative hypothesis, the distribution  $P(J|d, l_d = l)$  equals  $P(J|d)$  given by Equation 9, setting  $l_d$  to equal the suspected locus  $l$ . When assuming no disease locus, the distribution  $P(J|d, l_d = \infty)$  equals  $P(J|d)$  from Equation 9 under the assumption  $l_d = \infty$ .

We denote the standardized sums of  $S_n$  according to hypotheses  $H_0$  and  $H_1$  by  $S_n^{H_0}$  and  $S_n^{H_1}$ , respectively. The likelihood ratio under the two hypothesis can now be approximated as follows

$$\mathcal{L} = \frac{P(S_n|H_1)}{P(S_n|H_0)} \rightarrow \frac{P(S_n^{H_1})}{P(S_n^{H_0})} = \Lambda \quad (11)$$

The AAA method is defined to be the process of computing the LOD score  $\log_{10} \Lambda$  via Equation 11 at examined locations along the genome, declaring a region that shows a LOD above 3.3 as a suspect area that may contain a disease locus. Subsequently, significant peaks serve as candidates for fine-mapping. Section 3 details the process of selecting the LOD threshold.

## 2.4 Pooling Strategies

In the case of DNA pooling, two parameters affect the number of panels used, namely the pool size  $k$  and the number of pool repetitions  $l$ . It was shown that these two parameters can increase the accuracy of allele frequency estimation in the pooled sample which affects the method’s statistical power [13]. Based on previous studies, when using a high-throughput platform for genotyping, pooling is recommended to be applied in quadruplets ( $l = 4$ ). An empirical study of pooling examined the efficiency of this approach in association studies, using pools of  $k = 250$  individuals [15]. We report our results with  $l = 4$  and  $k = 200$ .

## 2.5 Leave-One-Out Filter

The leave-one-out (LOO) approach is a common filtering method that can be used in this context to discard false-positive signals originating from markers with erroneous frequencies. One potential source for bias is the inaccurate estimation of the allele frequencies in the ancestral populations. Biased genotyping errors can also result in false signals. Both error sources are assumed to occur independently between the markers and with low probability. When applying the AAA method, the robustness of a high LOD signal is examined via LOO by repeatedly removing markers and evaluating the effect on the LOD; the minimal LOD is reported, conferring with a conservative approach. A significant signal that persists after the removal of the marker with the highest contribution to the LOD is less likely to be false. LOO is especially effective in admixture mapping because suspected regions are usually supported by multiple SNP markers, retaining the method’s power throughout the filtering phase as opposed to association studies, which often pinpoints a small suspected region with a single SNP marker.

## 3 Results

In this section we evaluate the performance of AAA through simulations, showing that the method has high statistical power and can detect loci of disease genes with even modest ethnicity relative risk. We investigate our statistics in the absence of a disease, bounding the false-positive rate to 5% genome-wide. We examine the effect of deviation from model assumptions, showing that for many

realistic disease models ( $ERR > 1.4$ ) the method is robust to the inaccuracies expected in real data, and for milder ERR, the power can be retained through additional samples. We compare our AAA method to non-pooled MALD, demonstrating significant reduction in panel assays due to pooling at the cost of an increase in sample size. Finally, we validate our method by replicating the result of a prostate cancer risk locus using real data.

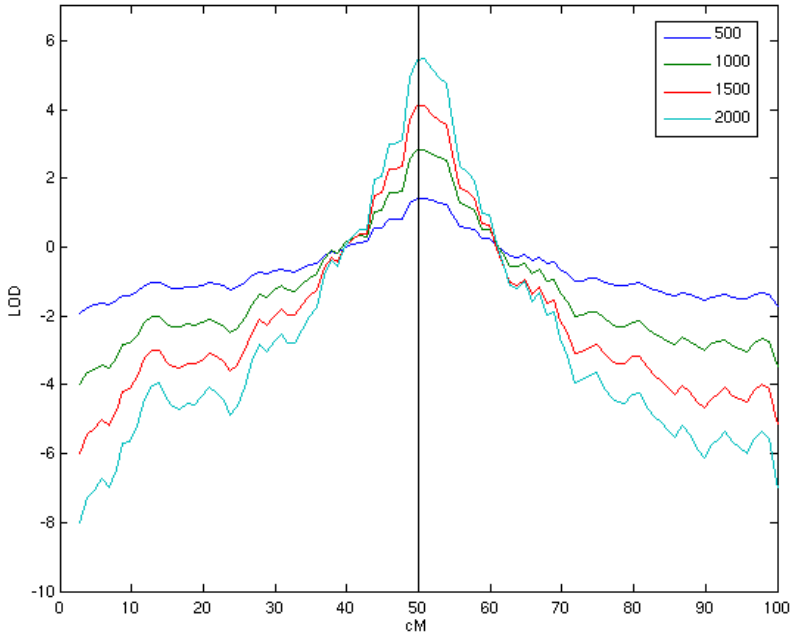
To evaluate the performance of our proposed method, we simulated data following the characteristics of recent MALD studies. We examined a range of disease models, including a mild value of  $ERR = 1.3$  (corresponding to end-stage kidney diseases) which produces signals that are harder to detect in comparison to diseases with higher ERR values such as hypertension ( $ERR 1.6$ ) [2]. The population of African Americans was simulated using the haplotypes of 60 unrelated European and 60 unrelated West African individuals phased in the HapMap project [22].

The simulation assumed a Hybrid-Isolated admixture model with 0.2 European contribution, 0.8 African contribution, and 8 generations of admixture. The simulated individuals were sampled according to a published panel of 1955 ancestry informative SNP markers [7], of which approximately 150 SNPs are on chromosome 1.

Figure 2 illustrates the output of the AAA method, using pools of 500, 1000, 1500 and 2000 affected individuals. The disease susceptibility locus was set to  $50cM$  and the simulated disease ERR was 1.3. A 3-marker sliding window was used to examine chromosome 1. One can clearly note that the evident peak, co-located with the disease locus, becomes significantly differentiated from distant locations with every increase in sample size.

We evaluated the distribution of our LOD statistic in the absence of a disease by performing simulations of pools of 500, 1000 and 2000 admixed controls, analyzing the sample using a window of 2, 3 and 4 markers at  $1cM$  steps. We assume an ERR between 1.3 and 1.8 using a multiplicative increase risk model (Equation 1) with a higher prevalence in Africans. Each configuration was repeated 2500 times. The results illustrate that the gap between random and significant signals increases markedly with both the sample size and the window size (see Table 2 in the appendix for more details). The 95<sup>th</sup> percentile was approximately  $LOD = 3.3$  when a pool of 1000 individuals was analyzed using a window of 2-4 markers over the entire genome, assuming an ERR of 1.3. This means that by defining the significance threshold to be a  $LOD > 3.3$ , we consequently confer a less than 5% type I error under the unfavorable condition of a hard to detect disease. Our recommended threshold of  $LOD > 3.3$  is applicable for a wide range of parameters, as seen in Table 2, but can be relaxed depending on the admixture model and sample size, as can be determined through appropriate simulations.

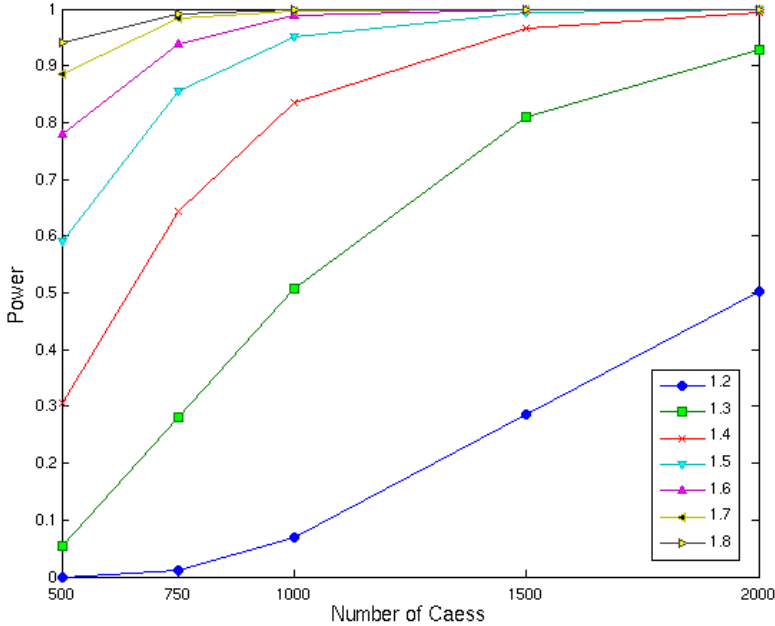
To establish the statistical power of AAA we simulated a range of models with ERR values ranging from 1.2 to 1.8. For each disease model, we evaluated the performance for a single pool of 500, 750, 1000, 1500 and 2000 cases. In each simulation, a uniformly random locus along chromosome 1 was chosen as the disease locus. Each configuration, consisting of a specific sample size and



**Fig. 2.** LOD score along chromosome 1 showing a peak co-located with disease locus at 50 cM. The significant signal is enhanced with the increase of sample size while nearby LOD scores drop. The simulated disease ERR (1.3) is comparable to end-stage kidney disease. Chromosome 1 was sampled using 147 ancestry informative markers.

an ethnicity relative risk, was repeated 2500 times. Figure 3 summarizes the results of applying AAA using a window of 4 markers. A successful detection was defined as a peak with  $\text{LOD} > 3.3$  within  $5\text{cM}$  of the actual disease locus. The results indicate high statistical power (over 80%) under disease models that are considered difficult to detect (e.g., ERR of 1.3) when a pool of 1500 affected individual is used. We further found that 500 cases suffice to detect a disease of  $\text{ERR} \geq 1.6$  with a power of approximately 80%, and 1000 cases yield a power of over 83% in the analysis of a disease with  $\text{ERR} \geq 1.4$ .

To evaluate the robustness of AAA to deviation from model assumptions, we examined the performance under inaccuracies in the admixture parameters. Namely, the inaccurate estimate  $\lambda$  of the number of generations since first admixture, and the inaccurate estimate of the ancestral distribution  $P(Q)$ . Using a simulated population with African American admixture characteristics we conclude that the statistical power is insensitive (less than 1% decrease in power) to an inaccuracy of up to 5% in  $\lambda$ . Error in the estimate of  $P(Q)$  has a greater effect on power. In particular, a 5% overestimation of the contribution of the ancestry with the higher risk yields a 4.8% drop in power for a study with 2000 cases and  $\text{ERR} \geq 1.5$ , and a 1.8% drop for a study with 1000 cases and  $\text{ERR} 1.8$ . When only 1000 cases are used to study a disease with a milder ERR of 1.5, the



**Fig. 3.** Statistical power of 4-marker window analysis under different disease models and sample sizes

power drops significantly from 95% to 72%. The inaccuracies in the estimation of these admixture parameters are expected to be lower than 5% in the case of African Americans [9].

To investigate the extent of genotyping reduction due to pooling we examined the number of SNP assays needed in order to achieve 70% power using our AAA method versus MALD. The MALD method performance was evaluated under the optimal condition where the ancestries are perfectly inferred by a fully informative single marker (as described in Section 2). The performance of AAA was examined over 2500 uniformly chosen locations along chromosome 1, using a window of 4 markers. In the case of AAA, we report the results under the configuration of  $k = 200$  and  $l = 4$  which resembles the choice of [13] and [15]. The results are shown in Table 1. For  $ERR = 1.3$ , MALD requires a sample of 700 affected individuals, with one assay per individual. For the same disease model, AAA uses 28 assays, which suggests a 96% reduction in genotyping. The disadvantage of AAA is the need to collect additional affected individuals. However, for less than doubling the number of individuals, a 25-fold reduction in the number of assays is achieved. The performance of AAA was evaluated using a real panel for admixture mapping. When considering only perfect markers, AAA performance improves even when a single marker window analysis is applied, reducing the number of cases from 1300 to 1200, and the number of assays from 28 to 24. Similar results are obtained for  $ERR = 1.4$ .

**Table 1.** The number of SNP assays needed to achieve a power of 70% using MALD and AAA. The AAA method yields over 25-fold decrease in the number of SNP assays when using pool size  $k = 200$  and number of replicates per pool  $l = 4$ .

ERR	Cases		Assays	
	MALD	AAA	MALD	AAA
1.3	700	1300	700	28
1.4	470	820	470	20

To evaluate the performance of AAA on real data, we examined a sample of 1646 African Americans with prostate cancer that were genotyped using 1985 ancestry informative SNPs. This sample led to the confirmation of prostate cancer risk locus in African American men through admixture mapping [3]. We simulated a pool using 600 cases that were genotyped with the same 1276 markers. The allele frequencies in the ancestral populations were estimated using a sample of 343 Europeans and 183 Africans. An ERR of 1.65 was used for the analysis based on [2]. A European genetic contribution of 0.215 was estimated using a maximum likelihood approach on the pooled sample of affected admixed individuals.

Applying AAA using a window of 4 markers results in a significant signal near a known risk locus (see Figure 4 in the appendix for more details). The peak on chromosome 8 (LOD 7.2) is less than 5Mb from the susceptibility locus reported by [3]. Applying the AAA method genome-wide yielded 2 additional less significant signals on chromosomes 5 and 9 (LOD 3.7 – 3.8). To evaluate the robustness of the three significant signals, we applied AAA with 4-marker and LOO filtering. The analysis shows that only the known locus on chromosome 8 persist, with a significant LOD of 5.88, while the other two peaks at chromosomes 5 and 9 drop to 0.2 and 1.46, respectively. We attribute the two additional signals to biased markers.

## 4 Discussion

Pool-based methods rely on estimates of the allele frequencies in the pooled sample. It is known that pool-based association analysis is sensitive to errors in these estimates. Previous studies evaluated an error in the estimation of allele frequency difference between pools of less than 1.4% in 10K SNP arrays [17]. We now discuss the effects of these errors on AAA.

The model we used to simulate allele frequencies assumed independent normally-distributed errors with zero mean. Three error levels were tested, adjusting the variance of the error so as to reflect a 95<sup>th</sup> percentile of 1, 3 and 5 percent error in observed allele frequency. We performed simulations using pools of 500, 1000 and 2000 admixed controls, analyzing a window of 4 markers while using LOO filtering at  $1cM$  steps, and assuming an ERR between 1.3 and 1.8. Each configuration was repeated 2500 times. The results are that the selected threshold of  $LOD = 3.3$  is still valid for up to 5% error in allele frequencies for the case of ERR 1.3 – 1.5 and

500 – 1000 affected individuals. These results further suggest that the analysis of the prostate cancer sample is robust to 5% allele frequency estimation error. Error in the estimation of allele frequencies has a greater impact on the false-positives rate in the case of a disease with a higher ERR or a larger sample, increasing the needed significance threshold defined by the 95<sup>th</sup> percentile. One should adjust the significance threshold according to the expected allele frequency error via appropriate simulation.

We also repeated the experiment with cases, evaluating the impact of allele frequency estimation errors on the statistical power of AAA. The power of analyzing a disease with ERR 1.5 using 1000 cases decreases from 95% to 82%. The tested error levels had a smaller effect on the analysis of a larger sample or a disease with a higher ERR value, still retaining a power of over 90%. In the analysis of a smaller sample size or a disease with a lower ERR, that achieved a power between 50% and 60% under accurate allele frequency estimation, the power decreased to 33 – 38% once such errors were introduced. However, in most of these settings, our simulated experiments on pooled controls suggest that a less stringent LOD threshold can be used without sacrificing the low level of false-positives.

The AAA method has an advantage over pooled association studies with respect to allele frequency estimation errors because (1) only a small fraction of SNP markers are required for the analysis, enabling the use of higher accuracy genotyping platforms, and (2) the chosen panel of markers are biased towards a high minor allele frequency in the admixed population, which increases the expected accuracy [18]. The common enhancements applied in pool-based association studies of repeated measures and the subdivision of samples into pools should also increase the robustness of our method considerably.

Another source of error lies in the inaccurate estimation of allele frequencies of the ancestral populations which may lead to an increase in the number of false-positive signals. Indeed, initial experiments indicate that errors in the ancestral allele distribution increase the false-positive signals as these mimic the effect of a true risk allele. Such results may explain few of the additional suspected regions in the prostate cancer sample that were detected prior to applying LOO.

Our analysis assumes knowledge of the admixture coefficient  $P(Q)$ , and the number of generations since the first admixture  $\lambda$ . While reasonable estimates of these parameters exists for some admixed populations, such as the African American and the Latino populations, it is recommended to tune the  $\lambda$  and  $P(Q)$  estimates using the sampled cases. We evaluated the genetic contribution of Europeans by applying a maximum likelihood approach on our prostate cancer cases pool, computing  $P(Q = \text{Europe}) = 0.215$ .

One of the properties of admixture mapping is that it can be applied on cases only, a property which holds for AAA as well. Nevertheless, similar to the use of control samples in MALD, healthy admixed individuals can increase the statistical power and decrease the rate of false-positives by providing a more accurate estimation of the allele frequencies in the ancestral population  $P(J|Q)$  as well as a more accurate estimation of the admixture parameters. Admixed controls pooled

in several groups, each of similar admixture coefficient, can be used to adjust the estimates of ancestral allele frequencies using a maximum likelihood approach. In particular, measuring a marker's frequency in two African American control groups with a known and different admixture coefficient allows the estimation of the marker's frequencies in the ancestral populations via Equation 5.

The AAA method presented in Section 2 is developed for the case of an admixed population that was formed by two ancestral populations. Supporting admixed populations with more than two ancestral populations, as is the case with the Latino admixed population who are descendants of Native Americans, Europeans, and Africans, can be achieved through an adjustment of Equation 7. Another approach is to model all low risk populations as one ancestral population and the high risk population as the second ancestral population, applying the method as is.

Our multi-marker AAA method takes into account knowledge of linkage-disequilibrium evident in the ancestral population. Such inherent and complete incorporation of LD in the analysis further increases the method's statistical power, whereas other MALD methods do not fully benefit from this information, ranging from partial to no support of background LD. In addition, the analysis we developed is applied on a window of markers, while common MALD statistics employ an analysis of a single locus. Interestingly, the development steps presented in Section 2 imply that non-pooled MALD methods can also benefit from a multi-marker approach by deriving a statistic that evaluates aberration of inferred ancestries in a region, examining a range of marker locations rather than a single marker location at a time.

The goal of this work has been to alleviate the considerable cost of mapping. As the results indicate, a high power of 70% can be achieved for a disease with ethnicity prevalence differences comparable with end-stage kidney disease by pooling 1300 affected individuals, yielding a 25-fold reduction in genotyping in comparison to previous non-pooled MALD methods. We showed that AAA can be used by gene mapping groups as an economical, practical and powerful approach for the initial localization of regions containing disease genes.

## Acknowledgement

The authors thank Karl Skorecky for fruitful discussions on the MALD method. The authors are grateful to David Reich for providing them with the SNP readings from his prostate cancer study which enabled the validation of the AAA method on real data. The authors thank Tamar Aizikowitz for her comments as well as help with our webpage design. Sivan Bercovici is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship. This research is partially supported by the Israel Science Foundation.

## Web Resources

The URL for AAAmap cited in the text is as follows: AAAmap Web site, <http://bioinfo.cs.technion.ac.il/AAAmap>.



## References

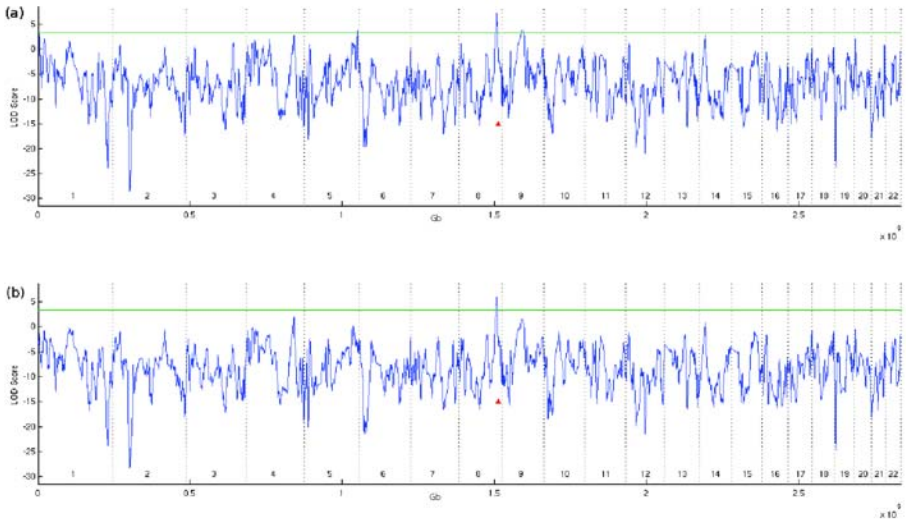
1. The Wellcome Trust Case Control Consortium: Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447(7145), 661–678 (2007)
2. Smith, M.W., O'Brien, S.J.: Mapping by admixture linkage disequilibrium: advances, limitations and guidelines. *Nat. Rev. Genet.* 6(8), 623–632 (2005)
3. Freedman, M.L., Haiman, C.A., Patterson, N., McDonald, G.J., Tandon, A., Waliszewska, A., Penney, K., Steen, R.G., Ardlie, K., John, E.M., Oakley-Girvan, I., Whittemore, A.S., Cooney, K.A., Ingles, S.A., Altshuler, D., Henderson, B.E., Reich, D.: Admixture mapping identifies 8q24 as a prostate cancer risk locus in African-American men. *Proceedings of the National Academy of Sciences* 103(38), 14068–14073 (2006)
4. Haiman, C.A., Patterson, N., Freedman, M.L., Myers, S.R., Pike, M.C., Waliszewska, A., Neubauer, J., Tandon, A., Schirmer, C., McDonald, G.J., Greenway, S.C., Stram, D.O., Le Marchand, L., Kolonel, L.N., Frasco, M., Wong, D., Pooler, L.C., Ardlie, K., Girvan, O.I., Whittemore, A.S., Cooney, K.A., John, E.M., Ingles, S.A., Altshuler, D., Henderson, B.E., Reich, D.: Multiple regions within 8q24 independently affect risk for prostate cancer. *Nat. Genet.* 39(5), 638–644 (2007)
5. Kao, W.H.L., Klag, M.J., Meoni, L.A., Reich, D., Berthier-Schaad, Y., Li, M., Coresh, J., Patterson, N., Tandon, A., Powe, N.R., Fink, N.E., Sadler, J.H., Weir, M.R., Abboud, H.E., Adler, S.G., Divers, J., Iyengar, S.K., Freedman, B.I., Kimmel, P.L., Knowler, W.C., Kohn, O.F., Kramp, K., Leehey, D.J., Nicholas, S.B., Pahl, M.V., Schelling, J.R., Sedor, J.R., Thornley-Brown, D., Winkler, C.A., Smith, M.W., Parekh, R.S.: Myh9 is associated with nondiabetic end-stage renal disease in african americans. *Nat. Genet.* 40(10), 1185–1192 (2008)
6. Smith, M.W., Patterson, N., Lautenberger, J.A., Truelove, A.L., McDonald, G.J., Waliszewska, A., Kessing, B.D., Malasky, M.J., Scafe, C., Le, E., De Jager, P.L., Mignault, A.A., Yi, Z., de Thé, G., Essex, M., Sankalé, J.L., Moore, J.H., Poku, K., Phair, J.P., Goedert, J.J., Vlahov, D., Williams, S.M., Tishkoff, S.A., Winkler, C.A., De La Vega, F.M., Woodage, T., Sninsky, J.J., Hafler, D.A., Altshuler, D., Gilbert, D.A., O'Brien, S.J., Reich, D.: A high-density admixture map for disease gene discovery in african americans 74(5), 1001–1013 (May 2004)
7. Bercovici, S., Geiger, D., Shlush, L., Skorecki, K., Templeton, A.: Panel construction for mapping in admixed populations via expected mutual information. *Genome Res.* 18(4), 661–667 (2008)
8. Reich, D., Patterson, N.: Will admixture mapping work to find disease genes? *Philosophical Transactions of the Royal Society B: Biological Sciences* 360(1460), 1605–1607 (2005)
9. Patterson, N., Hattangadi, N., Lane, B., Lohmueller, K.E., Hafler, D.A., Oksenberg, J.R., Hauser, S.L., Smith, M.W., O'Brien, S.J., Altshuler, D., Daly, M.J., Reich, D.: Methods for high-density admixture mapping of disease genes 74(5), 979–1000 (May 2004)
10. Tang, H., Coram, M., Wang, P., Zhu, X., Risch, N.: Reconstructing genetic ancestry blocks in admixed individuals (July 2006)
11. Bercovici, S., Geiger, D.: Inferring ancestries efficiently in admixed populations with linkage disequilibrium. *Journal of Computational Biology* 16(8), 1141–1150 (2009)

12. Chakraborty, R., Weiss, K.M.: Admixture as a tool for finding linked genes and detecting that difference from allelic association between loci. *Proceedings of the National Academy of Sciences of the United States of America* 85(23), 9119–9123 (1988)
13. Sham, P., Bader, J.S., Craig, I., O'Donovan, M., Owen, M.: Dna pooling: a tool for large-scale association studies. *Nat. Rev. Genet.* 3(11), 862–871 (2002)
14. Arnheim, N., Strange, C., Erlich, H.: Use of pooled dna samples to detect linkage disequilibrium of polymorphic restriction fragments and human disease: Studies of the hla class ii loci. *Proc. Natl. Acad. Sci.* 82(20), 6970–6974 (1985)
15. Steer, S., Abkevich, V., Gutin, A., Cordell, H.J., Gendall, K.L., Merriman, M.E., Rodger, R.A., Rowley, K.A., Chapman, P., Gow, P., Harrison, A.A., Highton, J., Jones, P.B.B., O'Donnell, J., Stamp, L., Fitzgerald, L., Iliev, D., Kouzmine, A., Tran, T., Skolnick, M.H., Timms, K.M., Lanchbury, J.S., Merriman, T.R.: Genomic dna pooling for whole-genome association scans in complex disease: empirical demonstration of efficacy in rheumatoid arthritis. *Genes Immun.* 8(1), 57–68 (2006)
16. Zeng, D., Lin, D.Y.: Estimating haplotype-disease associations with pooled genotype data. *Genetic epidemiology* 28(1), 70–82 (2005)
17. Kirov, G., Nikolov, I., Georgieva, L., Moskvina, V., Owen, M.J., O'donovan, M.C.: Pooled dna genotyping on affymetrix snp genotyping arrays. *BMC Genomics* 7(1) (February 2006)
18. Wilkening, S., Chen, B., Wirtenberger, M., Burwinkel, B., Forsti, A., Hemminki, K., Canzian, F.: Allelotyping of pooled dna with 250k snp microarrays. *BMC Genomics* 8, 77 (2007)
19. Darvasi, A., Soller, M.: Selective DNA Pooling for Determination of Linkage Between a Molecular Marker and a Quantitative Trait Locus. *Genetics* 138(4), 1365–1373 (1994)
20. Amundadottir, L.T., Sulem, P., Gudmundsson, J., Helgason, A., Baker, A., Agnarsson, B.A., Sigurdsson, A., Benediktsdottir, K.R., Cazier, J.B., Sainz, J., Jakobsdottir, M., Kostic, J., Magnusdottir, D.N., Ghosh, S., Agnarsson, K., Birgisdottir, B., Le Roux, L., Olafsdottir, A., Blondal, T., Andresdottir, M., Gretarsdottir, O.S., Bergthorsson, J.T., Gudbjartsson, D., Gylfason, A., Thorleifsson, G., Manolescu, A., Kristjansson, K., Geirsson, G., Isaksson, H., Douglas, J., Johansson, J.E., Bälter, K., Wiklund, F., Montie, J.E., Yu, X., Suarez, B.K., Ober, C., Cooney, K.A., Gronberg, H., Catalona, W.J., Einarsson, G.V., Barkardottir, R.B., Gulcher, J.R., Kong, A., Thorsteinsdottir, U., Stefansson, K.: A common variant associated with prostate cancer in european and african populations. *Nature Genetics* 38(6), 652–658 (2006)
21. Long, J.C.: The Genetic Structure of Admixed Populations. *Genetics* 127(2), 417–428 (1991)
22. The International HapMap Project.: A haplotype map of the human genome. *Nature* 437(7063), 1299–1320 (2005)

## Appendix

**Table 2.** The 95<sup>th</sup> percentile of LOD scores using pools of 500, 1000 and 2000 simulated controls analyzed using a window of 2, 3 and 4 markers under the false assumption of ERR between 1.3 and 1.8. All tested configurations exhibit a score lower than 3.3 in the 95<sup>th</sup> percentile. The simulations demonstrate that in most cases an increase in either sample size or in the size of the sliding window results in a reduction of the threshold.

Sample Size	2 Markers Window			3 Markers Window			4 Markers Window		
	1.3	1.5	1.8	1.3	1.5	1.8	1.3	1.5	1.8
500	2.8	3.28	3.12	2.83	3.28	2.99	2.84	3.26	2.72
1000	<b>3.29</b>	3.14	1.72	<b>3.3</b>	2.89	0.78	<b>3.28</b>	2.75	0.26
2000	3.28	1.56	-1.78	3.06	0.65	-4.4	2.93	0.05	-6.39



**Fig. 4.** The analysis of 600 prostate cancer cases using AAA and a 4 markers window. (a) The significant peak of 7.2 LOD is evident in close proximity to a validated prostate cancer risk locus at 129 Mb (marked by a triangle) that was previously discovered through a linkage scan by Amundadottir *et al.* [20] and later reported by Freedman *et al.* [3] using admixture mapping. Two additional significant signals are evident on chromosome 5 and 9. (b) Only the validated locus passes the LOO filter with a significant LOD of 5.88.

# Pathway-Based Functional Analysis of Metagenomes

Sivan Bercovici<sup>\*,\*\*</sup>, Itai Sharon<sup>\*,\*\*</sup>, Ron Y. Pinter, and Tomer Shlomi

Department of Computer Science, Technion, Haifa 32000, Israel  
{sberco, itaish, pinter, tomersh}@cs.technion.ac.il

**Abstract.** Metagenomic data enables the study of microbes and viruses through their DNA as retrieved directly from the environment in which they live. Functional analysis of metagenomes explores the abundance of gene families, pathways, and systems, rather than their taxonomy. Through such analysis researchers are able to identify those functional capabilities most important to organisms in the examined environment. Recently, a statistical framework for the functional analysis of metagenomes was described that focuses on gene families. Here we describe two pathway level computational models for functional analysis that take into account important, yet unaddressed issues such as pathway size, gene length and overlap in gene content among pathways. We test our models over carefully designed simulated data and propose novel approaches for performance evaluation. Our models significantly improve over current approach with respect to pathway ranking and the computations of relative abundance of pathways in environments.

**Keywords:** Metagenomics, functional analysis, pathways, Markov Chain Monte Carlo (MCMC).

## 1 Introduction

Metagenomics is an increasingly prevalent approach for the study of microbial communities directly from the environment in which they live. Unlike in traditional microbiology, random DNA pieces (called *reads*) – collected directly from the environment without a culturing stage – are being sequenced. Avoiding the culturing stage makes it possible to study the vast majority of microbes on earth, more than 99% according to some estimates, which cannot be cultured. To-date, metagenomics was applied for studying several environments and microbial functions [1-6]. Notable discoveries were made using metagenomics including the identification of proterhodopsin [7] and the discovery of photosystem I genes in viral genomes [8].

Analysis of metagenomic data poses analytical challenges resulting from the short length of DNA reads of which the data consists. Traditional Sanger sequencing generates reads of average length 900bps; newer high-throughput sequencers produce reads of even shorter lengths ranging between less than 100bps (*e.g.* the Illumina

---

\* These authors contributed equally to this work.

\*\* Corresponding authors.

Solexa and ABI SOLiD sequencers) to 500bps (the 454 Life Sciences sequencer). Even with recent and expected advances in sequencing technology, read length is likely to remain a major issue in metagenomics analysis that will require novel computational methods that are different from those used for the analysis of complete genomes. Such methods have been emerging in an increasing rate lately, including methods and strategies for assembly, gene calling, community structure prediction and more (see [9] for an overview of the field).

The functional analysis of metagenomes aims to identify those functional capabilities most significant to organisms living in the environment under study. Usually, analysis is done either at the single gene level, focusing on the abundance of gene families, or at the pathway level in which the occurrence of genes in pathways is taken into account. These processes start by identifying genes in the data and predicting their function, where function prediction is done by aligning the data against function-oriented databases. Such databases include COG [10], Pfam [11], and TIGRFAM [12] for gene level analysis, and KEGG [13], MetaCyc [14], or SEED [15] for systems or pathway level analysis<sup>1</sup>. For each function, the function prediction process generates its *read count*, *i.e.* the number of reads associated with the function in the metagenome. Once determined, read counts can be used for computing the relative abundance of each function in the metagenome. Previous works ignored issues related to gene length and the minimum portion of a gene required in order to identify it (*e.g.* [1, 16, 17]) and estimated the relative abundance of each function  $f$ , both at the gene family and pathway levels, as the relative abundance of its read count from all functions in the function database  $F$ :

$$freq(f) = \frac{read\_count(f)}{\sum_{f' \in F} read\_count(f')} \quad (1)$$

We refer to this as the *read count approach*. It is straightforward when complete genomes are considered and the relative abundance of functions is computed based on *gene count*, namely the number of genes associated with the different functions. However it results in inherently biased estimates when read counts are considered, due to the fact that longer genes are expected to have a higher read count simply due to their length. This problem is addressed in a recently published work [18] that presents a statistical framework for the functional analysis at the gene family level. The model presented in that paper is based on the assumption that the number of reads beginning at each position across any genome is Poisson-distributed [19]. While this framework fits gene families, it may not be suitable as is for functional analysis at the pathway level, most notably due to the presence of the same genes in several pathways.

Functional analysis at the pathway level is mainly used for two purposes: computation of pathway relative abundance, and pathway content comparison. Computing relative abundance of pathways within a single sample provides an overall view of the

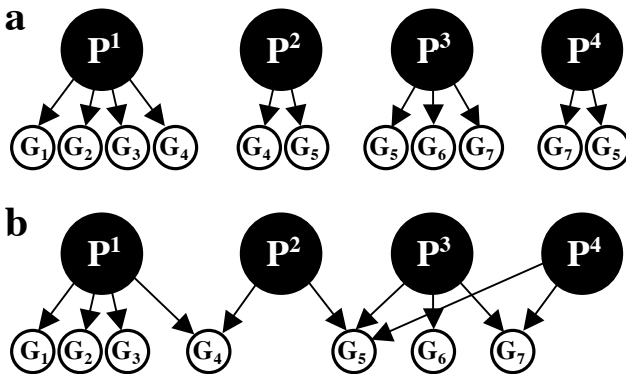
---

<sup>1</sup> The SEED database is commonly used for functional analysis that is defined in terms of subsystems rather than pathways. However, since our models treat pathways as gene sets and do not consider issues such as pathway topology and products, the theory described in this paper is also applicable to databases such as SEED.

environment and was used in many studies and platforms (e.g. [15, 17, 26, 28]). Comparing pathways' abundance between samples makes it possible to identify pathways that are enriched within one of the environments with respect to the other ([1,26]). Derivatives of pathway content comparison may be used for clustering functionally similar environments using metrics over pathway abundances vectors ([1,27]).

Pathway reconstruction is a related problem in which the most likely set of pathways in a genome or a metagenome is determined, without estimating their abundance. A commonly-used naïve approach to this problem would be to collect all pathways with at least one representative in the data. However, this approach is expected to yield an inflated list of pathways. Recently, a method called *MinPath* was described that attempts to deduce the minimal set of pathways required for supporting an observed set of functions [29]. The method uses Integer Programming for deciding whether a pathway is present, based on the observed functions. Note that in this case the relative abundance of the different functions is not taken into account, and no estimation of the relative abundance of the different pathways is done.

Here, we present two models for the functional analysis of metagenomes at the pathway level. Both models ignore pathway topology and treat pathways as gene sets. We begin with a short description of the model described in [18] and deduce the *independent pathways* model that can be regarded as a natural extension of the previous work. Next we present the *pathway intersection* model that takes into account the co-occurrences of genes in more than one pathway. We test both models on synthetic data and compare the results to the currently used read-count approach. Our tests focus on the abovementioned two common functional analysis tasks, namely sample comparison and the computation of relative abundance of pathways in the environment.



**Fig. 1. a. The independent pathways model.** In this model a gene that is shared among several pathways is assumed to have a copy for each pathway in which it appears. For example:  $G_5$  belongs to three pathways and thus assumed to have three copies **b. The pathway intersection model.** Each gene that appears in one or more pathways is assumed to appear once.  $G_5$  in this case will have a single copy, shared between  $P^2$ ,  $P^3$  and  $P^4$ .

## 2 Materials and Methods

### 2.1 The Poisson Model for Computing Gene Family Abundance

A metagenome  $M$  is a set of  $R$  sequence reads of length  $r$  each, extracted randomly with uniform probability for all positions across all genomes from some DNA sample of size  $L$  bps. A *gene family*  $G$  represents a set of functionally similar genes, which can be defined, for example, via sequence similarity. COG, Pfam and other databases are often used as references for the identification of gene families in metagenomic data. We denote a collection of gene families by  $D^{GENE}$ ; the association between  $M$ 's reads and gene families is defined in terms of the read count,  $R_G$ , representing the number of reads (out of  $R$ ) carrying a detectable portion of  $G$ 's member. Assuming that the abundance of a gene family  $G \in D^{GENE}$  in the DNA pool is  $C_G$  (i.e. the DNA sample has  $C_G$  copies of genes that are members in  $G$ ), the read count,  $R_G$ , is Poisson distributed with mean  $\lambda_G$  [18]:

$$\Pr(R_G = k) \sim \text{Poisson}(\lambda_G) = \frac{\lambda_G^k \cdot e^{-\lambda_G}}{k!} \quad (2)$$

where

$$\lambda_G = \frac{R}{L}(r + L_G - 2T) \cdot C_G \quad (3)$$

In this formula,  $R/L$  is the rate of read starts per base pair. The term  $(r + L_G - 2T)$  reflects the average number of starting positions for reads carrying a detectable portion of a single copy of  $G$  where  $L_G$  is the average length of  $G$ 's members,  $T$  is the minimum portion of a gene required to be present on a read in order to be associated with its family and  $r$  is the read length.

An estimator for a gene read count,  $\hat{R}_G$ , can be computed using BLAST [20] with a certain threshold. A Maximum Likelihood Estimate (MLE)  $\hat{C}_G$  for  $C_G$  can be calculated from Equation 3 and  $\hat{R}_G$ :

$$\hat{C}_G = \frac{\hat{R}_G}{\frac{R}{L} \cdot (r + L_G - 2T)} \quad (4)$$

All parameters in this formula are known, except for  $L$ , and hence an explicit calculation of gene family abundance is impossible. In previous work [18] the above formula was used to compute frequency estimators for gene families, which is the relative abundance of a certain gene family out of the total abundance of all gene families in the DNA sample pool (which eliminates the dependency on  $L$ )<sup>2</sup>. In this paper, we resolve the problem of the unknown DNA sample length  $L$  by computing the abundance of a gene family per organism in the sample, instead of the absolute abundance.

<sup>2</sup> Note that  $\lambda_G$  in [18] refers to the expected number of clone inserts whose two sides are sequenced. Here reads are assumed to be independent of each other, the adjustment to pair-end sequencing should be straightforward.

This requires an estimation of the average genome length in the DNA sample, as shown next.

## 2.2 Estimating the Average Genome Length in the DNA Sample

The estimation of the average length of a genome is based on the known existence of a group of genes that are known to be present exactly once per genome in all bacterial species. Several known single-copy genes, such as bacterial *rpoB*, *recA* and *gyrA*, were used as both phylogenetic markers [21, 22] as well as for the normalization of the abundance of genes in metagenomic samples [2, 22–25].

In the case of a single-copy gene *SCG*, the number of copies in the entire DNA sample,  $C_{SCG}$ , is equal to the number of organisms in the sample,  $N_0$ , hence it is possible to deduce an MLE for the average genome length based on Equation 4:

$$\frac{L}{N_0} \approx \frac{L}{\hat{C}_{SCG}} = \frac{R}{\hat{R}_{SCG}} (r + L_{SCG} - 2T) \quad (5)$$

A more accurate estimation of the average genome length is achieved by averaging the estimated values for several single copy genes.

Utilizing the estimated average genome length, based on Equation 4, the abundance of a gene family  $G$  per organism in the DNA sample can be calculated as following:

$$\frac{\hat{C}_G}{N_0} = \left( \frac{L}{N_0} \right) \cdot \frac{\hat{R}_G}{R \cdot (r + L_G - 2T)} \quad (6)$$

## 2.3 Computing Pathway Abundance: The Independent Pathways Model

In the context of the current analysis, a pathway  $P$  is defined as a set of gene families  $P = \{G_1^P, \dots, G_m^P\}$ ,  $G_i^P \subseteq D^{GENE}$ . Several repositories of pathways exist, for example KEGG and MetaCyc, and they can be used in this study. We denote a collection of pathways by  $D^{PATH}$ .

The *independent pathways model* assumes that all gene families within a certain pathway,  $P$ , have the same number of occurrences (*i.e.*  $C_1^P = C_2^P \dots = C_m^P$ ), and refer to this number of occurrences as the abundance of the pathway,  $C^P$ . In this section, we assume that pathways' abundances in an organism are mutually independent (Figure 1a). Analogously to the case of gene families, for each pathway  $P \in D^{PATH}$  our goal here is to compute the abundance of the pathway per organism, denoted by  $W^P$ . Based on the latter assumptions, for each pathway  $P$  an estimation of its abundance per organism can be calculated by averaging the estimated abundance of the member gene families:

$$W^P = \frac{\hat{C}^P}{N_0} = \left( \frac{L}{N_0} \right) \cdot \frac{1}{m} \sum_{i=1}^m \frac{\hat{R}_{G_i^P}}{R \cdot (r + L_{G_i^P} - 2T)} \quad (7)$$



Note that it is also possible to express the relative abundance of a pathway with respect to all other pathways in a sample by dividing  $W^p$  by the sum of  $W^i$  for all  $P^i \in D^{PATH}$ . In this case the estimation for the average genome length ( $L/N_0$ ) is eliminated.

## 2.4 Computing Pathway Abundance: The Pathways Intersection Model

Pathways – being a descriptive tool – are not necessarily disjoint modules, but rather they share common proteins. Ignoring the overlap in gene family content between pathways may lead the method of Section 2.3 to overestimate the abundance of pathways that share proteins with other pathways. Here we describe a second model that accounts for non-empty pathway intersections by jointly computing the abundance of all pathways within a collection of pathways.

The *pathways intersection model* assumes that a given pathway  $Y$  is either present or absent in an organism in the sample, where the presence of the pathway entails the presence of all of its member gene families in the organism. We denote by  $W^Y$  the random Boolean variable that represents the presence of a pathway  $Y$  in an organism. The probability that a gene family  $G$  is present in the genome of the organism is given by:

$$P(G|W) = 1 - \prod_{\{Y \in D^{PATH} | G \in Y\}} [1 - P(W^Y = 1)] \quad (8)$$

The abundance of  $G$  in the sample,  $C_G$ , is deduced by multiplying this probability by the number of organisms in the sample,  $N_0$ . Consequently the read count,  $R_G$ , is Poisson distributed with the following mean:

$$\lambda_G = \frac{R}{L} (r + L_G - 2T) \cdot \left( 1 - \prod_{\{Y \in D^{PATH} | G \in Y\}} [1 - P(W^Y = 1)] \right) \cdot N_0 \quad (9)$$

This can be computed for various estimates of the  $W$  variables, using the estimated average of the genome lengths in the sample (computed in Section 2.2).

Using the observed number of reads,  $\hat{R}_G$ , we estimate  $P(W^Y=1 | \hat{R}_G)$  via a Markov Chain Monte Carlo (MCMC) posterior sampling. We assume a uniform prior for  $P(W)$ , and estimate  $P(\hat{R}_G | W^Y=1)$  using Equation (2), with  $\lambda_G$  given by Equation (9). The average of the obtained samples is used as the estimated posterior probability for the presence of each pathway in an organism.

## 2.5 Materials

In order to test both our models we have generated five synthetic metagenomes based on simulated organisms, with different community complexities and metagenome sizes.

**Generating organisms.** We have generated two sets of organisms, KEGG10 and KEGG125 consisting of 10 and 125 synthetic species, respectively. First, the number of pathways and frequency of “dummy genes” (*i.e.* genes that do not belong to any

pathway and that were chosen at random) were chosen either manually (KEGG10) or at random using a normal distribution with manually set parameters (KEGG125). Next, the simulated number of pathways was chosen at random from the KEGG database. Having done that, all genes from the selected pathways and the dummy genes were placed at random on the genome, using lengths as they appear in KEGG (for enzymes) or 1000 (for dummy genes). In addition to these genes, 3 single copy genes, *gyrA*, *recA* and *rpoB*, were also located randomly on each genome using their true lengths, averaged over instances from several bacterial genomes (2670, 1040 and 3520 bps, respectively). Note that our simulated data is based on the pathway intersection model (of Section 2.4), namely a single copy for every gene that appears in at least one pathway. Overall, the average genome length, number of pathways and frequency of dummy genes was 2.5Mbps, 57 and 75% (respectively) for KEGG10 and 2.9Mbps, 81 and 68% for the KEGG125 (The choice of parameters was made in accordance with metagenomes in the IMG/M system [17]).

**Generating populations.** For each simulated population, a different organisms' prevalence and a different population structure were used. Population complexity, which refers to the relative abundance among species, was either high (similar abundance for most species) or low (a few relatively dominant species, low abundance for the rest).

**Metagenome generation.** Number of reads per metagenome was manually set, read length ( $r$ ) and minimum detectable gene portion ( $T$ ) were set to 900 (typical of Sanger sequencing [2]) and 100 (corresponds to e-value  $\approx 1e-100$  in BLAST) base-pairs, respectively. Number of reads per species is proportional to its DNA share in the population, defined as (genome length\*frequency in the population)/(sum of (genome length\*frequency in the population) over all species).

**Table 1.** General information on simulated metagenomes

Metagenome	Organisms	Population complexity (% of most abundant species)	# reads
M1	KEGG10	High (10%)	100,000
M2	KEGG10	Low (50%)	100,000
M3	KEGG125	High (1.4%)	100,000
M4	KEGG125	Low (10.8%)	100,000
M5	KEGG125	Low (10.8%)	10,000

## 2.6 Evaluation of the Different Methods

**Functional comparison.** In this test, the quality of each method with respect to pathway-based functional comparison of two samples is evaluated. Given two metagenomes  $M$  and  $M'$  and a method for pathway abundance estimation, the frequency of each pathway in both  $M$  and  $M'$  is estimated and the absolute difference between the two frequencies is computed. Next, pathways are ranked based on their differential enrichment, and the intersection between the true and estimated most differentially enriched pathways is computed for every prefix size  $m$  ( $\leq 100$ ).

**Pathway abundance estimations.** For each method, pathways are ranked based on their estimated frequencies. Similarly to the case of functional comparison, we use the number of pathways that are common to both the true and estimated  $m$  most abundant pathways as a measure of quality. Hyper-geometric distribution was used in order to evaluate the statistical significance of the results. In short, the probability that the intersection between the two lists of size  $m$  contains exactly  $k$  pathways is given by

$$\Pr(X = k) \sim \text{Hypergeometric}(k; N, m, n) = \frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}} \quad (10)$$

where  $N$  is the total number of pathways and  $n=m$  is the prefix size. The significance of the observed  $k$  is given by the Hyper-Geometric Tail (HGT):

$$\Pr(X \geq k) = \sum_{i=k}^m \text{Hypergeometric}(i; N, m, n) \quad (11)$$

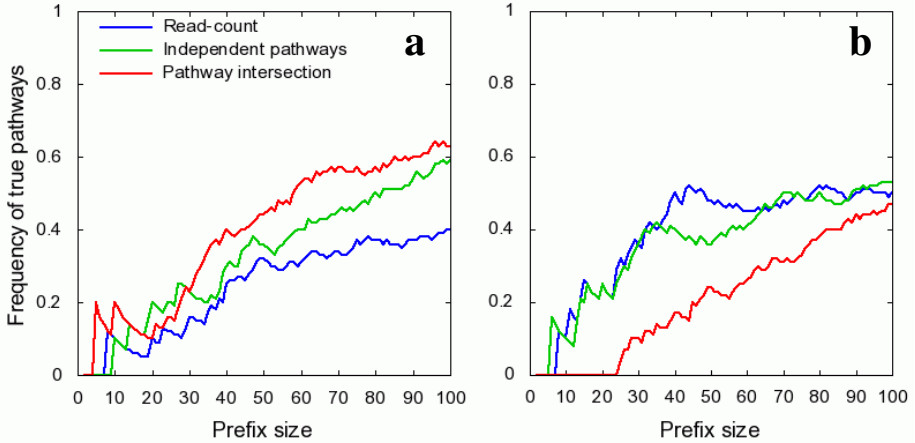
In addition to the above, we have also used the Pearson correlation coefficient for evaluating the degree of agreement between the lists of true and estimated frequencies.

### 3 Results

To evaluate the performance of our methods in predicting pathway abundances, we generated synthetic metagenome data with various community complexities and sizes (Section 2.5; Table 1). Our tests focus on two of the most interesting tasks in the context of metagenomics: (i) comparing pathways' abundance between samples and (ii) computing relative abundance of pathways within a single sample. As a baseline, we compared the performance of our methods to that of a standard read-count approach, estimating the relative abundance of each pathway as the relative abundance of its read counts out of the total number of read counts in all considered pathways (see Equation 1).

To evaluate the performance of the various prediction methods on the task of function comparison, we compared all pairs of metagenomes and evaluated the resulting lists (Figure 2, and Figure 5 in the Appendix). The pathway intersection model showed superior performance over the other models in 6 out of 10 scenarios (e.g. Figure 2a). The independent pathways model performed slightly better than the read-count model in most cases. The relatively low improvement in performance in this task is somewhat expected since our models aim to correct biases in the estimation of pathway abundances introduced by differences in pathway size and gene lengths, while these biases are largely eliminated when comparing the same pathway over two samples.

Since the original aim of the read-count method was to address the above task of computing changes in gene set abundances across metagenomes, it is not suitable for computing the relative abundance of pathways in a sample as it does not account for differences in pathway sizes (an inherent factor for this task). Therefore, as a baseline



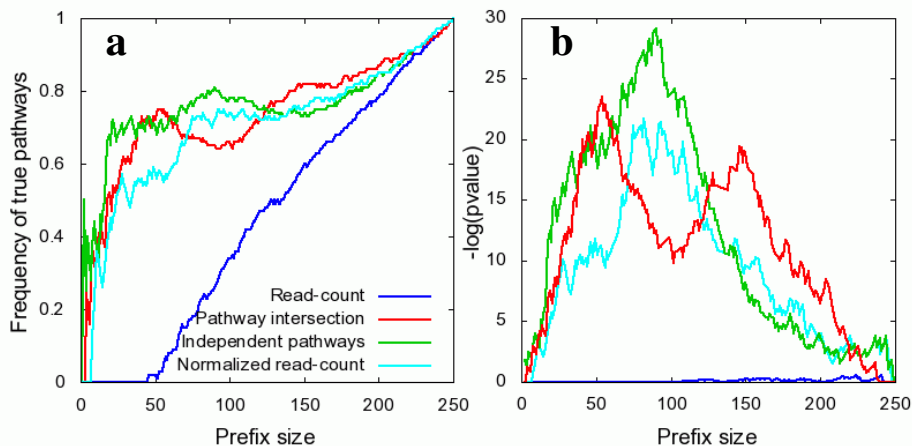
**Fig. 2. Agreement between true and estimated lists of most differentially enriched pathways in selected pairs of metagenomes.** For each  $m \leq 100$  (X axis), the frequency of pathways that are among the  $m$  most differentially enriched pathways in both the true and estimated ranked lists is plotted for (a) metagenomes M1 and M5, and (b) M3 and M4. Refer to the appendix for plots of all other pairs.

to assessing the performance of our methods here, we implemented a fourth method, the *normalized read-count*, that is based on read-counts but also account for pathway sizes. The relative abundance of a pathway in this method is given by:

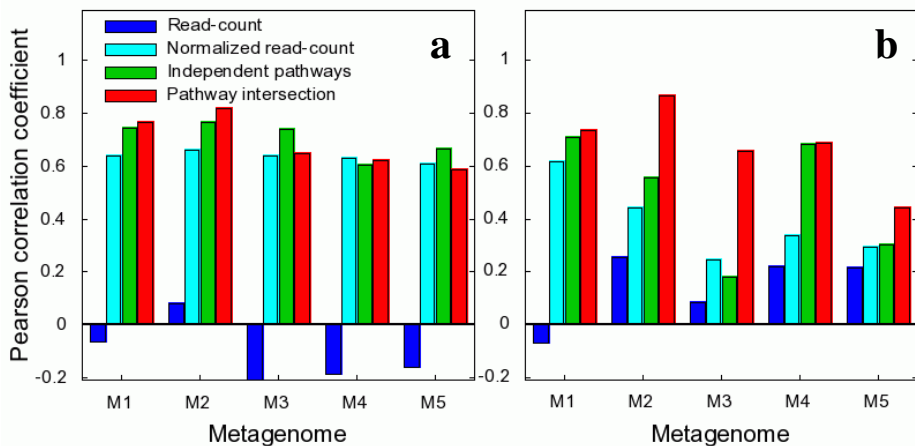
$$freq(P) = \frac{read\_count(P)/size(P)}{\sum_{P' \in D^{PATH}} read\_count(P')/size(P')} \quad (12)$$

To evaluate the performance of the various methods in predicting the relative abundance of pathways across a single sample, we tested the agreement between the rankings of pathways based on their true abundances and predicted abundances by the various methods (Figure 3, and Figure 6 in the Appendix). Quite expectedly, the performance of the read-count method is significantly worse than that of the other methods. A significant improvement is achieved by the normalized read-counts, taking into account pathway sizes. An additional marked improvement is achieved by the independent pathway model, accounting for variation in gene lengths. The performance of the pathway intersection method is inferior to that of the independent pathways method when ranking sets of highly abundant pathways. On the other hand, the performance of the pathway intersection method is superior to all other methods when considering pathways with lower abundances.

To further evaluate the performance of the various methods in predicting relative abundance of pathways across a single sample, we computed the Pearson correlation coefficient between the true and predicted relative abundances (Figure 4a). Consistent with its poor performance in the ranking test, the read-counts method shows no correlation with the true pathway abundances across all metagenomes. The independent pathways and the pathway intersection methods perform better than or equal to the



**Fig. 3. Ranking pathways based on their enrichment in metagenome M3.** (a) The intersection between the true and predicted  $m$  most abundant pathways using the various prediction methods (y-axis), for different values of  $m$  (x-axis). (b) Statistical significance (hypergeometric p-values) for the intersection between the true and predicted highly abundant pathway sets shown in (a). Other simulated metagenomes exhibited similar behavior (see Figure 6).



**Fig. 4. Pearson correlation between predicted and true pathway abundances across the various metagenomes.** (a) Correlations obtained for the entire set of 250 pathways used in the simulation, (b) correlations obtained for the set of 150 less abundant pathways.

normalized read counts method in all cases. In particular, the pathway intersection method outperforms the other approaches when lowly abundant pathways are considered (Figure 4b), as also shown above in the ranking tests. The success of the pathway intersection method on rare (or missing) pathways may be due to the fact that it does not do multiple counting of a gene common to several pathways while the other methods do. Frequencies assigned by the other models will be higher than the

true frequencies; while this also happens with abundant pathways, its influence on rare pathways is much higher. The independent pathways model does not suffer from this bias.

## 4 Discussion

In this work we have proposed two models for functional analysis of metagenomes at the pathway (systems) level reflecting two different assumptions regarding the sharing of genes among pathways. The two models eliminate biases resulting from variations in number of genes across pathways and also biases resulting from variation in genes' lengths [18]. Our methods performed much better with respect to predicting relative abundance of pathways. Each of our two methods was shown to have its own strength: the pathway intersection method outperforms the other approaches in predicting pathway abundances when focusing on lowly abundant pathways; the independent pathways method is superior in ranking pathway abundances for highly abundant pathways. Both our methods performed only slightly better than the read-count method when used for functional comparison, despite the failure of the later in the second task of predicting the absolute frequencies of the different pathways. One possible explanation for this behavior is that frequency estimation biases of specific pathways tend to be similar in both compared datasets and thus cancel each other when computing the relative abundances. For example, the relative abundance of a gene family or a pathway whose members are relatively long is likely to be overestimated by the read-count method in both samples. Such mutual compensation does not hold in the general case, suggesting that a more robust method is in place.

The pathway intersection method relies on the availability of single copy genes that are present in the vast majority of species in the studied environment. Single copy genes were used in the past as phylogenetic markers [3] and for estimating gene abundance [2, 23, 24]. There are several families of single copy genes that are known to be present across all known bacterial species, but these families are not present in Archaea and Eukaryotes. Hence, the pathway intersection method is more appropriate for environments in which the vast majority of sampled microbes are bacteria such as marine environments, but is likely to yield skewed frequencies when applied to environments in which either Archaeal or Eukaryotic species are abundant (such as acid mine drainage).

Functional characterization of metagenomic data such as that discussed in this study depends, first and foremost, on the quality of the employed pathway annotation data. Specifically, all pathway analysis methods rely on the basic assumption that a pathway is a coherent functional module that is either entirely present or absent in an organism. However, pathways defined in databases such as KEGG and MetaCyc do not fully address this requirement, and in many cases have only a fraction of their genes actually present in many species. Future advances in pathways curation are expected to significantly improve the outcome of the presented methods.

To our knowledge, this is the first time in which the issue of functional analysis at the pathway level of metagenomic data was studied in depth, providing further means for the exploration of metagenomes and their functions via environment-based comparative analysis.

**Acknowledgements.** Sivan Bercovici is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

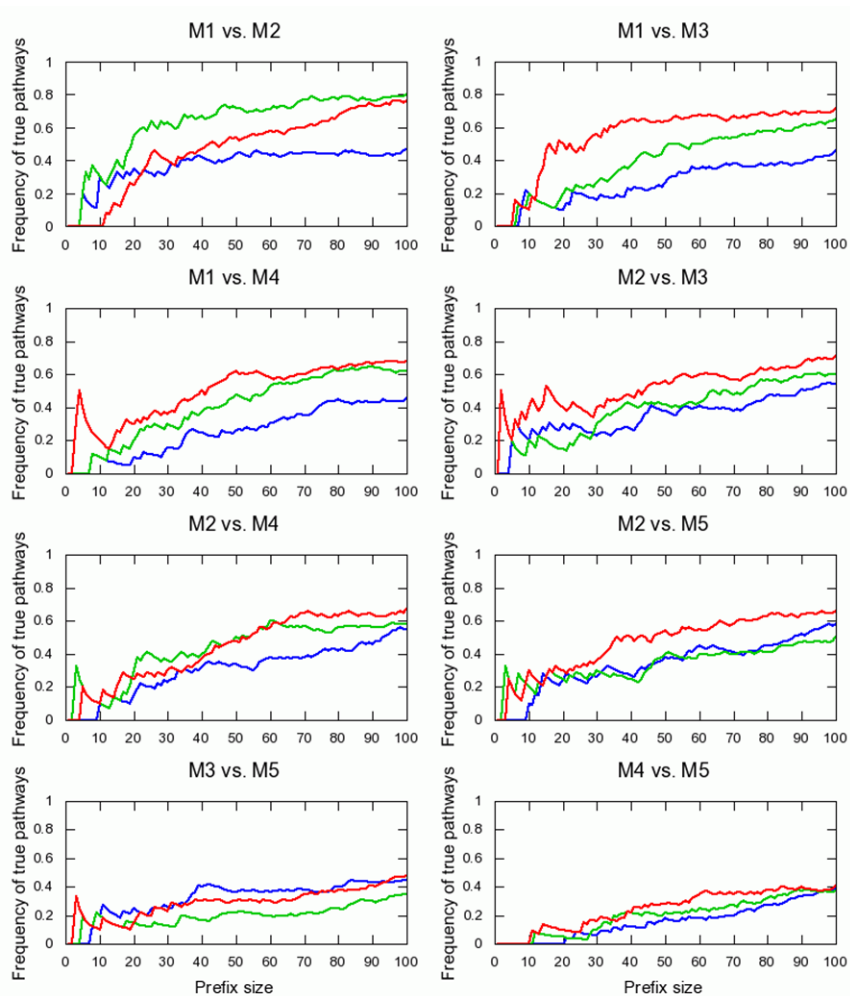
## References

1. DeLong, E.F., Preston, C.M., Mincer, T., Rich, V., Hallam, S.J., Frigaard, N., Martinez, A., Sullivan, M.B., Edwards, R., Brito, B.R., Chisholm, S.W., Karl, D.M.: Community Genomics Among Stratified Microbial Assemblages in the Ocean's Interior. *Science* 311(5760), 496–503 (2006)
2. Rusch, D.B., Halpern, A.L., Sutton, G., Heidelberg, K.B., Williamson, S., et al.: The Sorcerer II Global Ocean Sampling expedition: northwest Atlantic through eastern tropical Pacific. *PLoS Biol.* 5(3), e77 (2007)
3. Yooseph, S., Sutton, G., Rusch, D.B., Halpern, A.L., Williamson, S.J., et al.: The Sorcerer II Global Ocean Sampling Expedition: Expanding the Universe of Protein Families. *PLoS Biol.* 5(3), e16 (2007)
4. Gill, S.R., Pop, M., Deboy, R.T., Eckburg, P.B., Turnbaugh, P.J., Samuel, B.S., Gordon, J.I., Relman, D.A., Fraser-Liggett, C.M., Nelson, K.E.: Metagenomic Analysis of the Human Distal Gut Microbiome. *Science* 312(5778), 1355–1359 (2006)
5. Warnecke, F., Luginbuhl, P., Ivanova, N., Ghassemian, M., Richardson, T.H., et al.: Metagenomic and functional analysis of hindgut microbiota of a wood feeding higher termite. *Nature* 450, 560–565 (2007)
6. Tyson, G.W., Chapman, J., Hugenholtz, P., Allen, E.E., Ram, R.J., et al.: Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428(6978), 37–43 (2004)
7. Bèjà, O., Aravind, L., Koonin, E.V., Suzuki, M.T., Hadd, A., Nguyen, L.P., Jovanovich, S.B., Gates, C.M., Feldman, R.A., Spudich, J.L., Spudich, E.N., DeLong, E.F.: Bacterial rhodopsin: evidence for a new type of phototrophy in the sea. *Science* 289(5486), 1902–1906 (2000)
8. Sharon, I., Alperovitch, A., Rohwer, F., Haynes, M., Glaser, F., et al.: Photosystem-I gene cassettes are present in marine virus genomes. *Nature* 461, 258–262 (2009)
9. Raes, J., Foerstner, K.U., Bork, P.: Get the most out of your metagenome: computational analysis of environmental sequence data. *Curr. Opin. Microbiol.* 10(5), 490–498 (2007)
10. Tatusov, R.L., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Kiryutin, B., et al.: The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 4, 41 (2003)
11. Finn, R.D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S.R., Sonnhammer, E.L.L., Bateman, A.: Pfam: clans, web tools and services. *Nucleic Acids Res.* 34(Database Issue), D247–D251 (2006)
12. Haft, D.H., Selengut, J.D., White, O.: The TIGRFAMs database of protein families. *Nucleic Acids Res.* 31, 371–373 (2003)
13. Kanehisa, M., Goto, S.: KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* 28, 27–30 (2000)
14. Caspi, R., Foerster, H., Fulcher, C.A., Kaipa, P., Krummenacker, M., et al.: The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res.* 36(Database issue), D623–D631 (2008)
15. Overbeek, R., Begley, T., Butler, R.M., Choudhuri, J.V., Chuang, H.Y., et al.: The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Res.* 33, 5691–5702 (2005)

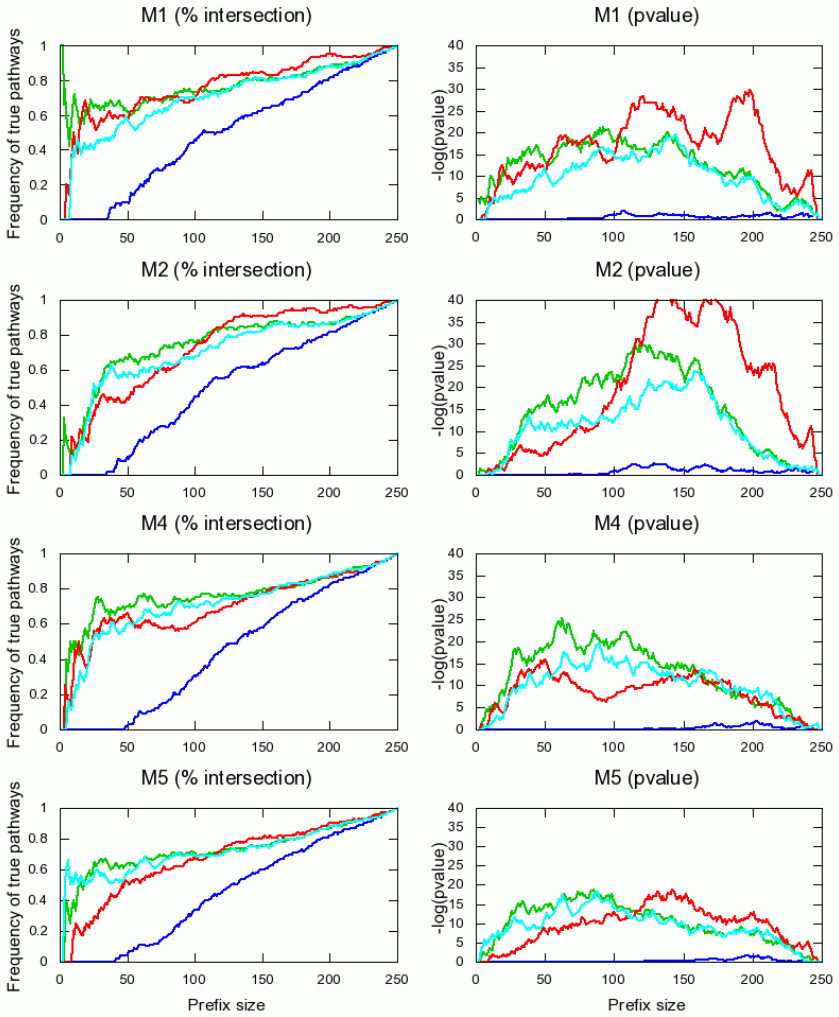
16. Rodriguez-Brito, B., Rohwer, F., Edwards, R.A.: An application of statistics to comparative metagenomics. *BMC Bioinformatics* 20(7), 162 (2006)
17. Markowitz, V.M., Szeto, E., Palaniappan, K., Grechkin, Y., Chu, K., Chen, I.A., Dubchak, I., Anderson, I., Lykidis, A., Mavromatis, K., Ivanova, N.N., Kyrpides, N.C.: The integrated microbial genomes (IMG) system in 2007: data content and analysis tool extensions. *Nucleic Acids Res.* 36(Database Issue), D528–D533 (2008)
18. Sharon, I., Pati, A., Markowitz, V.M., Pinter, R.Y.: A statistical framework for the functional analysis of metagenomes. In: Batzoglou, S. (ed.) *RECOMB 2009. LNCS*, vol. 5541, pp. 496–511. Springer, Heidelberg (2009)
19. Lander, E.S., Waterman, M.S.: Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* 2(3), 231–239 (1988)
20. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410 (1990)
21. Mollet, C., Drancourt, M., Raoult, D.: rpoB sequence analysis as a novel basis for bacterial identification. *Mol. Microbiol.* 26(5), 1005–1011 (1997)
22. Venter, J.C., Remington, K., Heidelberg, J.F., Halpern, A.L., Rusch, D., et al.: Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304(5667), 66–74 (2004)
23. Loy, A., Duller, S., Baranyi, C., Mußmann, M., Ott, J., et al.: Reverse dissimilatory sulfite reductase and other Dsr Proteins in sulfur-oxidizing bacteria: evolutionary history and suitability as phylogenetic markers. *Environ. Microbiol.* 11, 289–299 (2009)
24. Yutin, N., Suzuki, M.T., Teeling, H., Weber, M., Venter, J.C., et al.: Assessing diversity and biogeography of aerobic anoxygenic phototrophic bacteria in surface waters of the Atlantic and Pacific Oceans using the Global Ocean Sampling expedition metagenomes. *Environ. Microbiol.* 9, 1464–1475 (2007)
25. Howard, E.C., Henriksen, J.R., Buchan, A., Reisch, C.R., Bürgmann, H., et al.: Bacterial taxa that limit sulfur flux from the ocean. *Science* 314(5799), 649–652 (2006)
26. Edwards, R.A., Rodriguez-Brito, B., Wegley, L., Haynes, M., Breitbart, M., et al.: Using pyrosequencing to shed light on deep mine microbial ecology. *BMC Genomics* 7, 57 (2006)
27. Feingersch, R., Suzuki, M.T., Shmoish, M., Sharon, I., Sabehi, G., et al.: Microbial community genomics in eastern Mediterranean Sea surface waters. *ISME J.* (2009) doi:10.1038/ismej.2009.92
28. Dinsdale, E.A., Edwards, R.A., Hall, D., Angly, F., Breitbart, M., et al.: Functional metagenomic profiling of nine biomes. *Nature* 452, 629–632 (2008)
29. Ye, Y., Doak, T.G.: A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. *PLoS Comput. Biol.* 5(8), e1000465 (2009)



## Appendix



**Fig. 5. Agreement between true and estimated lists of most differentially enriched pairs of metagenomes.** Refer to the legend of Fig. 2 for description. Read count (blue), independent pathways (green) and pathway intersection (red) models are compared.




**Fig. 6. Ranking pathways based on their abundances in metagenomes M1, M2, M4 and M5.** Refer to the legend of Fig. 3 for description. Read count (blue), normalized read-count (cyan), independent pathways (green) and pathway intersection (red) models are compared.

# Hierarchical Generative Biclustering for MicroRNA Expression Analysis

José Caldas and Samuel Kaski

Aalto University School of Science and Technology  
Department of Information and Computer Science  
Helsinki Institute for Information Technology  
P.O. Box 15400, FI-00076 Aalto, Finland  
`jose.caldas@tkk.fi`, `samuel.kaski@tkk.fi`

**Abstract.** Clustering methods are a useful and common first step in gene expression studies, but the results may be hard to interpret. We bring in explicitly an indicator of which genes tie each cluster, changing the setup to biclustering. Furthermore, we make the indicators hierarchical, resulting in a hierarchy of progressively more specific biclusters. A non-parametric Bayesian formulation makes the model rigorous and yet flexible, and computations feasible. The formulation additionally offers a natural information retrieval relevance measure that allows relating samples in a principled manner. We show that the model outperforms other four biclustering procedures in a large miRNA data set. We also demonstrate the model’s added interpretability and information retrieval capability in a case study that highlights the potential and novel role of miR-224 in the association between melanoma and non-Hodgkin lymphoma. Software is publicly available. 

**Keywords:** Biclustering, graphical model, information retrieval, nested Chinese restaurant process, miRNA, melanoma, non-Hodgkin lymphoma.

## 1 Introduction

Unsupervised learning methods are often used as a first step in biological gene expression studies [\[1\]](#). The fact that most methods do not provide interpretable structures as to why the data was grouped as such hinders the subsequent analysis. Biclustering, where objects are both grouped and associated with feature subsets, is a natural framework for improving interpretability [\[2\]](#). Although several biclustering approaches exist, few are capable of handling the uncertainty that necessarily arises for a large enough number of biclusters. We recur to the probabilistic modelling framework [\[3\]](#) in order to develop a biclustering method that is interpretable, has flexibility and expressive power, and is efficiently computable. Probabilistic approaches to biclustering in the biological sciences have already been successfully used in the analysis of chemogenomic studies [\[4\]](#) and gene expression data [\[5\]](#), although the corresponding models differ significantly

---

<sup>1</sup> <http://www.cis.hut.fi/projects/mi/software/treebic/>

from ours. In particular, we propose a method to jointly group microarray samples hierarchically and assign genes to nodes in the hierarchy, with the node assignments implying that samples under the scope of a node in the hierarchy are homogeneous with respect to the genes assigned to it.<sup>2</sup> This enables the method to both provide a tree-structured clustering and explicitly state which features in the data were responsible for the groupings.

We show how the model yields a natural information retrieval relevance measure that allows relating samples in a principled manner. We apply the model to a large miRNA data set [6], compare it to other biclustering approaches, and illustrate the model’s advantages with a case study about the role of miR-224 on the relation between melanoma and non-Hodgkin lymphoma.

The paper is organized as follows: We first describe the model, its inference procedure, and an information retrieval relevance measure. We then compare our model to four other biclustering approaches in a miRNA data set, quantify the model’s information retrieval performance, and elaborate on a case study. Finally, we summarize our work and describe potential future directions.

## 2 Generative Model

### 2.1 Specification

The research problem is to find a hierarchy of clusters such that the objects (microarray samples) associated with a cluster are homogeneous for a subset of features (genes). Child clusters are to be associated with less objects but wider feature subsets than their corresponding parent clusters.

The proposed model can be seen as a particular instance of a biclustering method [2], where each bicluster corresponds to a group of samples that behave like replicates for a subset of genes. Biclusters are arranged as nodes in a tree hierarchy, with nodes closer to the root corresponding to broad sample groups tied by a low number of genes, and with nodes closer to the bottom of the hierarchy corresponding to limited but highly homogeneous sample groups. The generative process for our model consists of three parts: First, samples are partitioned into a tree structure. Second, genes are positioned along nodes in the tree. Third, the expression data is generated accordingly.

In order to partition samples into a tree structure, we use a probability distribution over infinitely-branched trees called the nested Chinese restaurant process (nCRP) [7]. This process may be defined over infinite-depth or finite-depth trees. We opt for specifying a maximum depth parameter in advance. Running the nCRP with a set of samples results in each sample being assigned a unique path from the root to a leaf node. The tree is initialized with a single node (the root), to which all samples are assigned. The samples are then probabilistically partitioned into groups according to the Chinese restaurant process (CRP) [8].<sup>3</sup>

<sup>2</sup> Alternatively, it may hierarchically group genes and assign samples to nodes, although we did not explore that option in the present work.

<sup>3</sup> Using the standard gastronomic metaphor associated with the CRP, we will interchangeably refer to groups as tables and samples as clients.

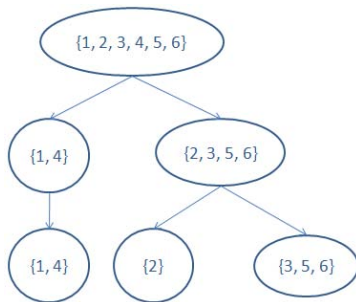
Formally, assume  $n$  clients are partitioned into  $k$  different tables ( $k \leq n$ ), making each of those tables  $j$  contain  $m_j$  clients. The assignment probabilities for the  $(n + 1)$ -th client are given as follows:

$$P(c_{n+1} = j | c_{1,\dots,n}) = \begin{cases} \frac{m_j}{n + \gamma}, & j \leq k, \\ \frac{\gamma}{n + \gamma}, & j = k + 1. \end{cases} \quad (1)$$

The joint distribution for all clients is exchangeable (i.e. invariant to client order permutation), with  $\gamma$  controlling the final number of tables. We consider  $\gamma$  to be a random variable with a vague prior distribution,

$$\gamma \sim \text{Gamma}(a_\gamma = 1, b_\gamma = 1). \quad (2)$$

The obtained tables become the child nodes of the root. The CRP is again run for each of the child nodes and corresponding clients. This recursion continues until the maximum tree depth has been reached. See Fig. [1](#) for an example.



**Fig. 1.** Running the nCRP for a set of 6 clients (numbered from 1 to 6) in an infinitely-branched tree of maximum depth 3. The clients assigned to each node are between braces.

Given the assignment of samples to paths in the tree, we represent genes as binary features and provide a feature activation model. First, for each directed edge  $(u, v)$  in the tree, we sample an edge length from a uniform Beta distribution,

$$l_{(u,v)} \sim \text{Beta}(\alpha = 1, \beta = 1). \quad (3)$$

All features (i.e. all genes) are set to 0 at the root node. For each directed edge from a node  $u$  to one of its child nodes  $v$ , each feature may switch to 1 with probability equal to the corresponding edge length. Finally, whenever a feature switches to 1, it stays at 1 for the remainder of the directed path. More formally, let  $z_{j,u}$  denote the value of feature  $j$  at node  $u$ . The activation of feature  $j$  at child node  $v$  is determined by the following conditional probabilities:

$$P(z_{j,v} = 1 | z_{j,u} = 0, l_{(u,v)}) = l_{(u,v)}, \quad (4)$$

$$P(z_{j,v} = 1 | z_{j,u} = 1, l_{(u,v)}) = 1. \quad (5)$$

This models the notion that genes may be indicative of either broad or specific phenotypes. By allowing genes to be activated along different paths, the model also encompasses the idea that two sample groups may be homogeneous with regard to the same gene, albeit in different ways, as we shall see below in more detail. Notice that the above probability rules are defined without recurring to assignments of samples to paths, that is, they can be formally defined as being applied on the entire infinite tree. The probability rule in (5) is also a component of the phylogenetic Indian buffet process (pIBP) model [9]. The scope of the two models is however disparate, as in the pIBP the authors present a non-exchangeable prior for representing objects as infinite feature vectors, where object relations are given in the form of a pre-specified tree.

The path assignment and feature activation patterns determine the distribution of the expression data. Assume that feature  $j$  switches from 0 to 1 at node  $u$ . Denote the set of samples in the subtree that has  $u$  as its root by  $S_u$ , and the expression data for those samples restricted to feature  $j$  as  $\mathbf{Y}_{j,S_u}$ . Then,

$$\mathbf{Y}_{j,S_u} \sim N(\mu_{j,u}\mathbf{1}, \sigma_{j,u}^2\mathbf{I}), \quad (6)$$

$$\mu_{j,u} \sim N(\mu = 0, \sigma^2 = \sigma_{j,u}^2), \quad (7)$$

$$\sigma_{j,u}^2 \sim \text{Inv-Gamma}(a = 1, b = 1). \quad (8)$$

where  $\mu_{j,u}$  and  $\sigma_{j,u}^2$  are respectively scalar mean and variance parameters, specific to the group induced by feature  $j$  at node  $u$ . The prior distribution for each  $\mu_{j,u}$  assumes adequately normalized data; the random variable  $\sigma_{j,u}^2$  is given a vague prior distribution. Our choice of prior probability density functions allows us to analytically integrate out  $\mu_{j,u}$  and  $\sigma_{j,u}^2$ , obtaining a multivariate Student- $t$  distribution for  $\mathbf{Y}_{j,S_u}$  [10]. This increases the efficiency of the sampler, although normality assumptions are in practice only an approximation whose usefulness is ultimately only validated by the results. If, for a given path ending in a leaf node  $u$ , a feature  $j$  never becomes activated, then, for every sample  $s \in S_u$ , we draw the corresponding scalar expression value  $Y_{j,s}$  from a baseline Gaussian distribution, assuming standardized data,

$$Y_{j,s} \sim N(\mu_0 = 0, \sigma_0^2 = 1). \quad (9)$$

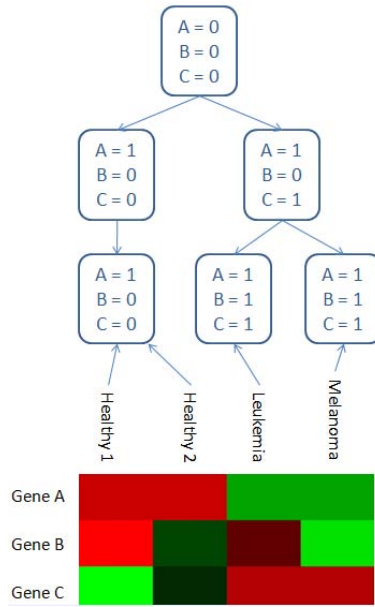
Notice that the same baseline distribution is used when required, regardless of the actual path or feature.

Figure 2 provides an example of an artificial data set with 3 genes and 4 samples generated using this approach.

## 2.2 Inference

We are interested in analyzing the joint posterior distribution of the path assignment and feature activation variables (respectively,  $\mathbf{c}$  and  $\mathbf{z}$ ), as well as  $\gamma$ , given the input expression data, which according to Bayes' rule is

$$P(\mathbf{c}, \mathbf{z}, \gamma | \mathbf{Y}) = \frac{P(\gamma)P(\mathbf{c}|\gamma)P(\mathbf{z})P(\mathbf{Y}|\mathbf{c}, \mathbf{z})}{P(\mathbf{Y})} \propto P(\gamma)P(\mathbf{c}|\gamma)P(\mathbf{z})P(\mathbf{Y}|\mathbf{c}, \mathbf{z}). \quad (10)$$



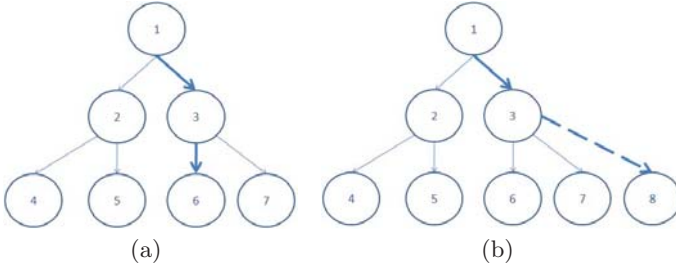
**Fig. 2.** Illustration of the generative process for a fictitious noise-free data set with 3 genes (A, B, and C) and 4 samples (“healthy 1”, “healthy 2”, “leukemia”, and “melanoma”). The healthy samples share the same path assignment, while each of the cancer samples has its own unique path. The rounded rectangles represent nodes and indicate the current feature activation state. Gene A becomes active at both of the root’s child nodes, leading to homogeneous expression for the healthy samples as well as for the cancer samples, although the between-group difference in expression is significant. Gene C exhibits homogeneous expression under both cancer samples, but not under the healthy samples. Gene B has a specific expression pattern for each of the samples.

The term  $P(\mathbf{z})$  results from integrating out all edge length variables, and the term  $P(\mathbf{Y}|\mathbf{c}, \mathbf{z})$  results from integrating out all mean and variance variables. The posterior distribution is intractable and we approximate it by means of a collapsed Gibbs sampler [11, 12].

**Sampling path assignments.** The posterior distribution for the path assignment of client  $i$  is given by

$$P(c_i|\mathbf{c}_{-i}, \mathbf{z}, \mathbf{Y}) \propto P(c_i|\mathbf{c}_{-i})P(\mathbf{Y}_{\cdot, i}|\mathbf{c}, \mathbf{z}, \mathbf{Y}_{\cdot, -i}), \quad (11)$$

where  $\mathbf{c}_{-i}$  is the collection of path assignments for all clients except  $i$ ,  $\mathbf{Y}_{\cdot, -i}$  is the expression data for all features and all clients but  $i$ , and dependency on  $\gamma$  has been dropped from the notation for succinctness. See Fig. 3 for an illustration of path assignments. The number of available paths to choose from is equal to the total number of nodes in the current tree (discarding any previous path



**Fig. 3.** Two possible paths for a new client, given a tree with 7 nodes. In [3\(a\)](#), the path 1→3→6 does not involve the creation of new nodes. In [3\(b\)](#), the path 1→3→8 implies adding a new node to the tree.

assignments of client  $i$ ). The first term in [\(11\)](#) can be computed with [\(1\)](#). The second term can be decomposed into the following product:

$$P(\mathbf{Y}_{\cdot,i}|\mathbf{c}, \mathbf{z}, \mathbf{Y}_{\cdot,-i}) = \left( \prod_{l=1}^L \prod_{j=1}^G P(Y_{j,i}|\mathbf{Y}_{j,S_{u_l}})^{z_{j,u_l}(1-z_{j,p(u_l)})} \right) \prod_{j=1}^G P(Y_{j,i})^{1-z_{j,u_L}}. \tag{12}$$

Each node  $u_l$  corresponds to the node at the  $l$ -th level on the given path. We denote the parent of  $u$  by  $p(u)$ . The first term in [\(12\)](#) is interpretable as follows: For every node  $u_l$  in the path, we take the features that switch to 1 in that node. For each of those features  $j$ , we consider the clients assigned to paths that include  $u_l$ , and compute the predictive probability of the corresponding induced group generating the observed  $Y_{j,i}$ . It is straightforward to derive that the predictive distribution for each induced group is a univariate Student- $t$  distribution [\[10\]](#). The second term in [\(12\)](#) involves the features that are never activated in the path. For each of those, we must compute the probability that  $Y_{j,i}$  was generated from a baseline Gaussian distribution, as described in [\(9\)](#). For every previously unpopulated section of a path, there needs to be an instantiation of the corresponding feature activation variables. We choose to draw them from their prior distribution. Since feature values are formally generated throughout the entire infinite tree, our approach conceptually corresponds to a type of *lazy loading*, where feature values are instantiated from their prior distribution as required. This implies that, although we are effectively bringing in novel feature variables into the model, their specific values do not contribute to the probability computations in [\(11\)](#). Alternative approaches involving simultaneously sampling path assignments and novel feature values are however possible.

**Sampling feature values.** The posterior odds for the value of feature  $j$  at node  $u$  are given by

$$\frac{P(z_{j,u} = 1|\mathbf{c}, \mathbf{z}_{-(j,u)}, \mathbf{Y})}{P(z_{j,u} = 0|\mathbf{c}, \mathbf{z}_{-(j,u)}, \mathbf{Y})} = \frac{P(z_{j,u} = 1|\mathbf{z}_{-(j,u)}) P(\mathbf{Y}_{j,\cdot}|z_{j,u} = 1, \mathbf{z}_{-(j,u)}, \mathbf{c})}{P(z_{j,u} = 0|\mathbf{z}_{-(j,u)}) P(\mathbf{Y}_{j,\cdot}|z_{j,u} = 0, \mathbf{z}_{-(j,u)}, \mathbf{c})}, \tag{13}$$



where  $\mathbf{z}_{-(j,u)}$  is the set of feature values excluding feature  $j$  at node  $u$ ,  $\mathbf{Y}_{j,\cdot}$  is the expression data restricted to feature  $j$ , and  $\mathbf{z}_{j,\cdot}$  is the set of feature values for all nodes but restricted to feature  $j$ . Due to the conditional probability distributions specified in (4) and (5), some feature values are deterministic and thus do not require sampling. Namely, if a feature is set to 1 at a node  $u$ , then all values for that feature at any node  $v$  descendant from  $u$  must be equal to 1. This entails that the process of sampling a feature value corresponds to incrementing or decrementing the feature's generality level for a specific path.

The first term in (13) is given by

$$\frac{P(z_{j,u} = 1 | \mathbf{z}_{-(j,u)})}{P(z_{j,u} = 0 | \mathbf{z}_{-(j,u)})} = \frac{\alpha + n_{u+}^{-j}}{\beta + n_{u-}^{-j}}, \quad (14)$$

where  $n_{u+}^{-j}$  is the number of features that switched from 0 to 1 when traversing the edge  $(w, u)$  ( $w$  being the parent node of  $u$ ) and  $n_{u-}^{-j}$  is the number of features that were kept at 0 when traversing that same edge, with both parameters disregarding feature  $j$ . The second term in (13) is given by

$$\frac{P(\mathbf{Y}_{j,\cdot} | z_{j,u} = 1, \mathbf{z}_{-(j,u)}, \mathbf{c})}{P(\mathbf{Y}_{j,\cdot} | z_{j,u} = 0, \mathbf{z}_{-(j,u)}, \mathbf{c})} = \frac{P(\mathbf{Y}_{j,S_u} | z_{j,u} = 1, \mathbf{c})}{\prod_{i=1}^{d_u} P(\mathbf{Y}_{j,S_{v_i}} | z_{j,v_i} = 1, z_{j,u} = 0, \mathbf{c})}, \quad (15)$$

where  $v_i$  is the  $i$ -th child node of  $u$ , and  $d_u$  is the total number of child nodes of  $u$ . Both the numerator and the terms in the denominator correspond to multivariate Student- $t$  distributions. In the numerator, all samples under node  $u$  are assumed to form a group with respect to feature  $j$ . In the denominator, samples instead form subgroups, each of them homogeneous with respect to feature  $j$ , but without assuming between-group homogeneity.

**Sampling  $\gamma$ .** We sample the variable  $\gamma$  by use of an auxiliary variable scheme developed for Dirichlet process mixture models [13,14]. The procedure presented here is identical to the one in the hierarchical Dirichlet process model [14]. For a given node  $u$  in the tree, let  $d_u$  be the number of its child nodes and  $n_u$  be the number of samples assigned to it. It can be shown [15] that  $d_u$  is distributed as

$$P(d_u | \gamma, n_u) \propto \gamma^{d_u} \frac{\Gamma(\gamma)}{\Gamma(\gamma + n_u)}, \quad (16)$$

where terms that do not depend on  $\gamma$  have been discarded. Multiplying the above over all nodes in the tree yields

$$P(d_1, \dots, d_V | \gamma, n_1, \dots, n_V) \propto \prod_{u=1}^V \gamma^{d_u} \frac{\Gamma(\gamma)}{\Gamma(\gamma + n_u)}, \quad (17)$$

where the product is taken across all nodes  $u$  that are not leaf nodes, and  $V$  designates the number of those nodes. The posterior distribution for  $\gamma$  depends exclusively on its prior distribution from (2) and the above product. The main

idea behind this sampling scheme is to represent each fraction of Gamma functions as

$$\frac{\Gamma(\gamma)}{\Gamma(\gamma + n_u)} = \frac{1}{\Gamma(n_u)} \int_0^1 w_u^\gamma (1 - w_u)^{n_u - 1} \left(1 + \frac{n_u}{\gamma}\right) dw_u, \quad (18)$$

where  $w_u \in [0, 1]$  is an auxiliary variable. Define  $\mathbf{w} = (w_u)_{u=1}^V$ , introduce an extra vector of binary auxiliary variables  $\mathbf{b} = (b_u)_{u=1}^V$ , and specify the joint distribution of  $\gamma$ ,  $\mathbf{w}$ , and  $\mathbf{b}$  as

$$q(\gamma, \mathbf{w}, \mathbf{b}) \propto \gamma^{a_\gamma - 1 + d} \cdot e^{-\gamma b_\gamma} \prod_{u=1}^V w_u^\gamma (1 - w_u)^{n_u - 1} \left(\frac{n_u}{\gamma}\right)^{b_u}, \quad (19)$$

where we have used dot ( $\cdot$ ) notation for vector summation. Marginalizing the auxiliary variables from the above joint distribution yields the original posterior distribution for  $\gamma$  [14, 13]. Gibbs sampling updates are then given by

$$q(\gamma | \mathbf{w}, \mathbf{b}) \propto \gamma^{a_\gamma - 1 + d} \cdot e^{-\gamma(b_\gamma - \sum_{u=1}^V \log w_u)}, \quad (20)$$

$$q(w_u | \gamma) \propto w_u^\gamma (1 - w_u)^{n_u - 1}, \quad (21)$$

$$q(b_u | \gamma) \propto \left(\frac{n_u}{\gamma}\right)^{b_u}. \quad (22)$$

Visual inspection of the sampled values shows that the sampler converges under 50 iterations.

### 2.3 Information Retrieval

Generative models offer a natural measure of pairwise object relevance. Consider an arbitrary probabilistic model parameterized by  $\theta$  with input data  $\mathbf{X}$ . Assume a query object  $q$ , corresponding to the data point  $x_q$ , and a potentially relevant object  $r$ . Denote the parameters relating to  $r$  as  $\theta_r$ . The relevance of  $r$  to  $q$  can be defined as

$$rel(q, r) \stackrel{\text{def}}{=} \int_{\theta} P(x_q | \theta_r) P(\theta | X) d\theta \quad (23)$$

[16]. This measure can be interpreted as the expected probability that the data point corresponding to object  $q$  was generated with the parameters from object  $r$ . A standard approximation is to obtain an estimate  $\hat{\theta}$  and compute  $P(x_q | \hat{\theta}_r)$ . Notice that this measure is not symmetric.

In our context, the relevance of a sample  $r$  to another sample  $q$  can be defined as the expected probability that the expression data  $\mathbf{Y}_{\cdot, q}$  was generated with the path variable  $c_r$ . This implies that any two samples  $r_1$  and  $r_2$  with equal path assignments ( $c_{r_1} = c_{r_2}$ ) are equally relevant to a query sample  $q$ . Thus, in this model the proposed relevance measure works at node granularity. Averaging (23) over samples yields an estimate of between-node relevance, although we have not explored this possibility in the present work. We approximate (23) by using only the sample with the highest posterior probability, generated via the described Gibbs sampler.

### 3 Results

We tested our model on a collection of 199 miRNAs profiled in 218 human healthy tissues, tumors, and cell lines. We pre-processed the data set and standardized the resulting expression data in a gene-wise fashion, as originally described [6]; this makes the data set coherent with the parameter choices stipulated in the previous section. We ran the Gibbs sampler for 2500 burn-in iterations and further 2500 iterations, collecting the sample with the highest posterior probability. The path and feature variables were initialized with a draw from their prior. The maximum tree depth was fixed at 3, which is the lowest number that allows the model to form a sample hierarchy. The method took about 14 hours to run on an AMD Opteron Dual Core Processor with 2.8GHZ<sup>4</sup>. This procedure was repeated 30 times. In the following analysis, we considered the sample with the overall highest posterior probability.

#### 3.1 Comparison to Previous Work

We compared the performance of our method to that of 4 well-established biclustering approaches [17,18,19,20] with default parameterizations. The results are presented in table 1. As miRNAs are known to have tissue-specific expression profiles [21], we first tested for the enrichment of specific tissues in the obtained biclusters. Significance was computed by means of Bonferroni-corrected hypergeometric tests with an original p-value of 0.01. Our method, named TreeBic, had the highest fraction of biclusters enriched for at least one tissue; at the other extreme, the CC method failed to significantly cluster samples from the same tissues in any bicluster. Our method, along with Samba, also managed to obtain the highest number of tissues enriched in at least one bicluster. Next, we assessed the functional homogeneity of each bicluster. We extracted a collection of confirmed miRNA targets from the TarBase database [22]. For each bicluster, we took the corresponding miRNAs and obtained the union of their targets. We then computed the functional enrichment of Gene Ontology (GO) [23] biological process terms in each target set, again using a Bonferroni-corrected hypergeometric test with an original p-value of 0.01 (terms with 5 or less genes were discarded). Our method outperforms all others with respect to the number of enriched GO categories. The biclusters found by our method also appear to be overall more functionally homogeneous, as shown by the percentage of biclusters enriched for at least one GO category. The overall low number of enriched GO categories is possibly due to the current sparsity of confirmed microRNA targets. Despite these results, our method has the second-lowest number of biclusters.

---

<sup>4</sup> Preliminary experiments on an artificial data set with 218 samples and 5970 features indicate that the same simulation takes approximately 250 hours, with the average number of nodes in the inferred tree being 145. Path variable sampling takes approximately 95% of inference time, indicating that a combination of heterogeneous features and high sample size leading to a large tree is the main bottleneck in the method.

**Table 1.** Method comparison with regard to tissue and miRNA target gene functional enrichment. Our model is named TreeBic; it outperforms 4 standard biclustering methods both in the fraction of biclusters enriched for at least one tissue/GO category and in the total number of enriched tissues/GO categories. See text for details on the meaning of each performance measure.

	TreeBic	Samba <a href="#">[17]</a>	Plaid <a href="#">[18]</a>	CC <a href="#">[19]</a>	OPSM <a href="#">[20]</a>
# Biclusters	16	54	29	20	10
% Tissue-Enriched Biclusters	<b>63%</b>	50%	41%	0%	40%
% GO Term-Enriched Biclusters	<b>63%</b>	46%	0%	18%	60%
# Enriched Tissues	<b>14</b>	<b>14</b>	8	0	2
# Enriched GO Terms	<b>12</b>	11	0	4	9

This suggests that the inferred hierarchical structure allows for a more efficient representation of the signal in the data set. Overall, by performing best both in terms of the fraction of enriched biclusters and the total number of enriched tissue and GO categories, our method appears to dominate over the other tested approaches.

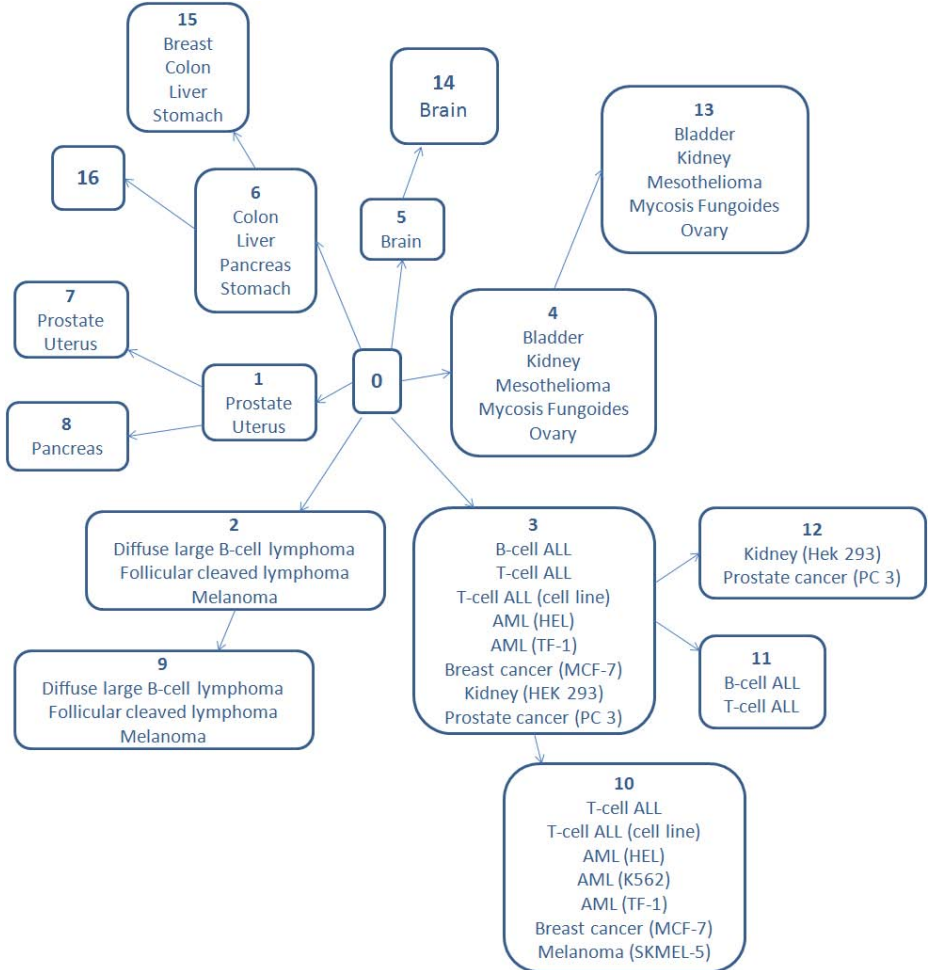
### 3.2 Information Retrieval

In previous work we have shown that graphical models are useful in deriving object relevance measures that allow performing information retrieval in gene expression data in both an efficient and interpretable manner [\[24\]](#). Here, we performed a feasibility study on the ability of the model to retrieve samples from the same tissue as a query sample. For a query taken from one of the 218 samples, we defined as positive the samples with the same tissue as the query. We computed true-positive and false-positive rates at each point in the relevance-ranked list of leaf nodes, and summarized the measures with the area under the corresponding ROC curve (AUC) [\[25\]](#). For each tissue or cell line class, we computed the median of the AUC. Out of 20 classes, 13 (65%) led to a median AUC higher than the 0.5 baseline. The list of ranked results can provide important biological insight. As a case study, we queried the system with a follicular cleaved lymphoma sample. The method considers a sequence of 2 melanoma samples, 7 follicular cleaved lymphoma samples, and 6 large B-cell lymphoma samples as the most relevant. Although melanoma is a malignancy of a different cell type than non-Hodgkin lymphoma, there is epidemiological evidence for their association [\[26\]](#), a relation which is highlighted in our results and which we further investigate below. The practical usefulness of this model for information retrieval remains to be further assessed.

### 3.3 Biological Analysis

Figure [4](#) portrays the inferred sample tree. The method separates samples into organs from the reproductive system (node 1, with the exception of ovary, which

falls under node 4), malignancies (nodes 2 and 3), and organs from the gastrointestinal tract (node 6). The method isolates the only two brain tissue samples in the data set, with a potential explanation being that they are the only healthy samples of ectodermal origin in the data set, in contrast with e.g. organs from node 6, which are of endodermal origin. On the other hand, node 4 appears to contain a more heterogeneous set of enriched tissues and pathological entities,



**Fig. 4.** Inferred tree structure. Nodes are numbered in breadth-first order and labelled with overrepresented tissues or cell lines (FDR  $q$ -value  $< 0.25$ ). The non-stringent  $q$ -value enables richer node annotations. Some of the tissue types are overrepresented in more than one leaf node (e.g. T-cell ALL in nodes 10 and 11). Notice that this annotation approach does not guarantee that significant tissues in a parent node are also significant in the corresponding child nodes (e.g. nodes 6 and 15). Node 16 did not have any significantly overrepresented tissues.

**Table 2.** Genes differentially over-expressed between two melanoma sample groups (designated as types A and B) [28]. Genes predicted to be miR-224 targets are in bold text.

Gene Function	Over-Expressed in Type A	Over-Expressed in Type B
Pro-Apoptotic	<b>APAF1</b> , BAD, BNIP1, <b>BNIP3L</b> , CASP1, <b>CASP7</b> , CYCS, <b>VDAC1</b>	BAK1, <b>CASP2</b> , CASP4, <b>ENDO</b> G, HTRA2, PDCD5, PRODH, SEPT4, TNFSF10
Anti-Apoptotic	<b>BCL2</b> , BCL2A1, PPAR $\delta$ , RAF1	<b>API5</b> , <b>FIS1</b> , PPP2CA, PPP2R1A, <b>PPP2R1B</b> , <b>PSEN1</b>
Antioxidant	GLRX2, GPX4, GSR, MT3, <b>PRDX3</b> , PRDX5	ATOX1, <b>CAT</b> , GSS, HSPD1, <b>SOD1</b>

including a combination of healthy (bladder, kidney, and ovary) and cancerous (mesothelioma, mycosis fungoides) tissues. The method is also able to further decompose leukemias (node 3) into leukemia cell lines (node 10) and leukemic tissue (node 11).

The previously mentioned relation between melanoma and non-Hodgkin lymphoma is also hinted at by the contents of node 2. In order to find miRNAs with a role specifically in both melanoma and lymphoma, we computed the set difference between miRNAs that are activated in the melanoma and lymphoma nodes and those which are activated in any of the other haematological malignancy nodes. The single resulting miRNA, miR-224, is known to have a dual function, conditionally inducing both apoptosis and cell proliferation, and it was found to be either over or under-expressed in several tumor types [27]. In order to grasp potential mechanisms by which miR-224 may have a common role in melanoma and lymphoma, we first analyzed a collection of 38 genes that were found to be differentially over-expressed between two subsets of melanoma samples in an independent study (designated as type A and B) [28]. We used a recent miRNA target prediction algorithm [29] to compute which of those genes are potential miR-224 targets (Table 2). The prediction that 50% of type-A pro-apoptotic genes and 67% of type-B anti-apoptotic genes are regulated by miR-224 is evidence of its dual role in cell proliferation and apoptosis, and indicative that it may have an important post-transcriptional regulatory effect in melanoma. The role of miR-224 in stimulating proliferation is not well understood [27]. We hypothesize that it may enhance proliferation by targeting some of the predicted type-A pro-apoptotic genes. The anti-apoptotic gene API-5, recently proposed as a target for cancer treatment [36], is known to be targeted by miR-224 [30], and its protein product interacts with FGF-2 [31], which has in turn been observed to have increased levels of expression in patients with haematological malignancies, including lymphoma [32]. There is also evidence that miR-224 directly binds CD40 [33], which is known to have an important role both in lymphoma [34] and melanoma [35]. Together, these results indicate miR-224 may be an important element in explaining the association between melanoma and non-Hodgkin lymphoma. Although this analysis is speculative, it brings out the model's ability to generate hypotheses and drive the biological analysis.

## 4 Conclusions

We have introduced a graphical model which allows grouping microarray samples and providing an interpretation basis for that grouping. The model makes the assumption that samples are grouped in a tree structure, where nodes correspond to hierarchical subgroups, and where each node is associated with a subset of genes for which the corresponding samples are highly homogeneous. We applied the model to a large miRNA data set, where it was shown to outperform other biclustering approaches. We then provided a case study that depicts how the model variables and information retrieval formulation can be used to direct the biological analysis. The case study highlighted the potential role of miR-224 in the association between melanoma and non-Hodgkin lymphoma.

The current model may be extended in several ways. While in the present work we fixed the maximum tree depth at a specific level, selection of the appropriate depth may be conducted by recurring to cross-validation measures or by enhancing the model with an automatic depth selection capability. The assumption that each sample chooses a single path allows for the use of a flexible prior over trees that also makes computations feasible. This assumption can be relaxed, although it may lead to slower mixing during inference. Finally, alternative feature activation models may be devised, incorporating notions such as e.g. pathway enrichment among genes activated throughout the same edges.

**Acknowledgments.** We thank Leo Lahti for helpful comments. This work was supported by the Finnish Funding Agency for Technology and Innovation (TEKES, grant no. 40101/07). J.C. and S.K. belong to the Finnish Centre of Excellence on Adaptive Informatics Research, and were additionally supported by the Pattern Analysis, Statistical Modelling and Computational Learning Network of Excellence (PASCAL 2 EU Network of Excellence, grant no. ICT 216886). J.C. is additionally supported by a doctoral grant from the Portuguese Science and Technology Foundation (FCT, grant no. SFRH/BD/35974/2007).

## References

1. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster Analysis and Display of Genome-Wide Expression Patterns. *P. Natl. Acad. Sci. U.S.A.* 95, 14863–14868 (1998)
2. Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 1, 24–45 (2004)
3. Jordan, M.I. (ed.): *Learning in Graphical Models*. MIT Press, Cambridge (1999)
4. Flaherty, P., et al.: A Latent Variable Model for Chemogenomic Profiling. *Bioinformatics* 21, 3286–3293 (2005)
5. Gerber, G.K., et al.: Automated Discovery of Functional Generality of Human Gene Expression Programs. *PLoS Comput. Biol.* 3, 1426–1440 (2007)
6. Lu, J., et al.: MicroRNA Expression Profiles Classify Human Cancers. *Nature* 435, 834–838 (2005)
7. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The Nested Chinese Restaurant Process and Bayesian Inference of Topic Hierarchies. *J. ACM* (to appear)



8. Aldous, D.: Exchangeability and Related Topics. In: *École d'été de probabilités de Saint-Flour, XIII*, pp. 1–198. Springer, Berlin (1985)
9. Miller, K.T., Griffiths, T.L., Jordan, M.I.: The Phylogenetic Indian Buffet Process: A Non-Exchangeable Nonparametric Prior for Latent Features. In: McAllester, D., Myllymaki, P. (eds.) *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pp. 403–410. AUAI Press, Corvallis (2008)
10. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: *Bayesian Data Analysis*, 2nd edn. Chapman & Hall/CRC, Boca Raton (2004)
11. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, Boca Raton (1996)
12. Liu, J.S.: The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *J. Am. Stat. Assoc.* 89, 958–966 (1994)
13. Escobar, M.D., West, M.: Bayesian Density Estimation and Inference Using Mixtures. *J. Am. Stat. Assoc.* 90, 577–588 (1995)
14. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet Processes. *J. Am. Stat. Assoc.* 101, 1566–1581 (2006)
15. Antoniak, C.E.: Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *Ann. Stat.* 2, 1152–1174 (1974)
16. Buntine, W., et al.: A Scalable Topic-Based Open Source Search Engine. In: Zhong, N., et al. (eds.) *Proceedings of the IEEE/WIC/ACM Conference on Web Intelligence*, pp. 228–234. IEEE Computer Society, Los Alamitos (2004)
17. Tanay, A., Sharan, R., Shamir, R.: Discovering Statistically Significant Biclusters in Gene Expression Data. *Bioinformatics* 18, S136–S144 (2002)
18. Lazzeroni, L., Owen, A.: Plaid Models for Gene Expression Data. *Stat. Sinica* 12, 61–86 (2002)
19. Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: Bourne, P., et al. (eds.) *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103. AAAI Press, Menlo Park (2000)
20. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. In: Istrail, S., Waterman, M.S., Clark, A.G. (eds.) *Proceedings of the Sixth Annual International Conference on Computational Biology*, pp. 49–57. ACM, New York (2002)
21. Landgraf, P., et al.: A Mammalian MicroRNA Expression Atlas Based on Small RNA Library Sequencing. *Cell* 129, 1401–1414 (2007)
22. Papadopoulos, G.L., Reczko, M., Simossis, V.A., Sethupathy, P., Hatzigeorgiou, A.G.: The Database of Experimentally Supported Targets: A Functional Update of TarBase. *Nucleic Acids Res.* 37, D155–D158 (2008)
23. Ashburner, M., et al.: Gene Ontology: Tool for the Unification of Biology. *Nat. Genet.* 25, 25–29 (2000)
24. Caldas, J., et al.: Probabilistic Retrieval and Visualization of Biologically Relevant Microarray Experiments. *Bioinformatics* 25, i145–i153 (2009)
25. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008)
26. Lens, M.B., Newton-Bishop, J.A.: An Association Between Cutaneous Melanoma and Non-Hodgkin's Lymphoma: Pooled Analysis of Published Data with a Review. *Ann. Oncol.* 16, 460–465 (2004)
27. Wang, Y., Lee, C.G.L.: MicroRNA and Cancer: Focus on Apoptosis. *J. Cell. Mol. Med.* 13, 12–23 (2009)
28. Su, D.M., et al.: Two Types of Human Malignant Melanoma Cell Lines Revealed by Expression Patterns of Mitochondrial and Survival-Apoptosis Genes: Implications for Malignant Melanoma Therapy. *Mol. Cancer Ther.* 8, 1292–1304 (2009)



29. Kertesz, M., et al.: The Role of Site Accessibility in MicroRNA Target Recognition. *Nat. Genet.* 39, 1278–1284 (2007)
30. Wang, Y., et al.: Profiling MicroRNA Expression in Hepatocellular Carcinoma Reveals MicroRNA-224 Up-regulation and Apoptosis Inhibitor-5 as a MicroRNA-224-specific Target. *J. Biol. Chem.* 283, 13205–13215 (2008)
31. Van den Berghe, L., et al.: FIF [Fibroblast Growth Factor-2 (FGF-2)-Interacting-Factor], a Nuclear Putatively Antiapoptotic Factor, Interacts Specifically with FGF-2. *Mol. Endocrinol.* 14, 1709–1724 (2000)
32. Krejci, P., et al.: FGF-2 Expression and its Action in Human Leukemia and Lymphoma Cell Lines. *Leukemia* 17, 817–819 (2002)
33. Mees, S.T., et al.: Involvement of CD40 Targeting Mir-224 and Mir-486 on the Progression of Pancreatic Ductal Adenocarcinomas. *Ann. Surg. Oncol.* 16, 2339–2350 (2009)
34. French, R.R., et al.: CD40 Antibody Evokes a Cytotoxic T-Cell Response that Eradicates Lymphoma and Bypasses T-Cell Help. *Nat. Med.* 5, 548–553 (1999)
35. Pirozzi, G., et al.: CD40 Expressed on Human Melanoma Cells Mediates T Cell Co-Stimulation and Tumor Cell Growth. *Int. Immunol.* 12, 787–795 (2000)
36. Rigou, P., et al.: The Antiapoptotic Protein AAC-11 Interacts with and Regulates Acinus-Mediated DNA Fragmentation. *EMBO J.* 28, 1576–1588 (2009)

# Subnetwork State Functions Define Dysregulated Subnetworks in Cancer

Salim A. Chowdhury<sup>1</sup>, Rod K. Nibbe<sup>2,4</sup>,  
Mark R. Chance<sup>3,4</sup>, and Mehmet Koyutürk<sup>1,4</sup>

<sup>1</sup> Dept. of Electrical Engineering & Computer Science

<sup>2</sup> Dept. of Pharmacology

<sup>3</sup> Dept. of Physiology & Biophysics

<sup>4</sup> Center for Proteomics & Bioinformatics

Case Western Reserve University, Cleveland, OH 44106, USA

{`sxc426,rkn6,mrc16,mxk331`}@case.edu

**Abstract.** Emerging research demonstrates the potential of protein-protein interaction (PPI) networks in uncovering the mechanistic bases of cancers, through identification of interacting proteins that are coordinately dysregulated in tumorigenic and metastatic samples. When used as features for classification, such coordinately dysregulated subnetworks improve diagnosis and prognosis of cancer considerably over single-gene markers. However, existing methods formulate coordination between multiple genes through additive representation of their expression profiles and utilize greedy heuristics to identify dysregulated subnetworks, which may not be well suited to the potentially combinatorial nature of coordinate dysregulation. Here, we propose a combinatorial formulation of coordinate dysregulation and decompose the resulting objective function to cast the problem as one of identifying subnetwork state functions that are indicative of phenotype. Based on this formulation, we show that coordinate dysregulation of larger subnetworks can be bounded using simple statistics on smaller subnetworks. We then use these bounds to devise an efficient algorithm, CRANE, that can search the subnetwork space more effectively than simple greedy algorithms. Comprehensive cross-classification experiments show that subnetworks identified by CRANE significantly outperform those identified by greedy algorithms in predicting metastasis of colorectal cancer (CRC).

## 1 Introduction

Recent advances in high-throughput screening techniques enable studies of complex phenotypes in terms of their associated molecular mechanisms. While genomic studies provide insights into genetic differences that relate to certain phenotypes, functional genomics (*e.g.*, gene expression, protein expression) helps elucidate the variation in the activity of cellular systems [1]. However, cellular systems are orchestrated through combinatorial organization of thousands of biomolecules [2]. This complexity is reflected in the diversity of phenotypic effects, which generally present themselves as weak signals in the expression

profiles of single molecules. For this reason, researchers increasingly focus on identification of multiple markers that together exhibit differential expression with respect to various phenotypes [3,4].

**Network-based approaches to identification of multiple markers.** High-throughput protein-protein interaction (PPI) data [5] provide an excellent substrate for network-based identification of multiple interacting markers. Network-based analyses of diverse phenotypes show that products of genes that are implicated in similar phenotypes are clustered together into “hot spots” in PPI networks [6,7]. This observation is exploited to identify novel genetic markers based on network connectivity [8,9,10]. For the identification of differentially expressed subnetworks with respect to GAL80 deletion in yeast, Ideker *et al.* [11] propose a method that is based on searching for connected subgraphs with high aggregate significance of individual differential expression. Variations of this method are shown to be effective in identifying multiple genetic markers in prostate cancer [12], melanoma [13], diabetes [14], and others [15,16,17].

**Coordinate/synergistic dysregulation.** Network-based approaches are further elaborated to capture coordinate dysregulation of interacting proteins at a sample-specific resolution [18]. Ulitksy *et al.* [19] define dysregulated pathways as subnetworks composed of products of genes that are dysregulated in a large fraction of phenotype samples. Chuang *et al.* [20] define subnetwork activity as the aggregate expression of genes in the subnetwork, quantify the dysregulation of a subnetwork in terms of the mutual information between subnetwork activity and phenotype, and develop greedy algorithms to identify subnetworks that exhibit significant dysregulation. Subnetworks identified by this approach are also used as features for classification of breast cancer metastasis, providing significant improvement over single-gene markers [20]. Nibbe *et al.* [21,22] show that this notion of coordinate dysregulation is also effective in integrating protein and mRNA expression data to identify important subnetworks in colon cancer (CRC). Anastassiou [23] introduces the concept of synergy to delineate the complementarity of multiple genes in the manifestation of phenotype. While identification of multiple genes with synergistic dysregulation is intractable [23], important insights can still be gained through pairwise assessment of synergy [24].

**Contributions of this study.** Despite significant advances, existing approaches to the identification of coordinately dysregulated subnetworks have important limitations, including the following: (i) additive formulation of subnetwork activity can only highlight the coordinate dysregulation of interacting proteins that are dysregulated in the same direction, overlooking the effects of inhibitory and other complex forms of interactions; (ii) greedy algorithms may not be able to adequately capture the coordination between multiple genes that provide weak individual signals. In this paper, with a view to addressing these challenges, we develop a novel algorithm, CRANE, for the identification of Combinatorially dysRegulAted subNEtworks. The contributions of the proposed computational framework include the following:

- We formulate coordinate dysregulation combinatorially, in terms of the mutual information between *subnetwork state functions* (specific combinations of quantized mRNA expression levels of proteins in a subnetwork) and phenotype (as opposed to additive *subnetwork activity*).
- We decompose combinatorial coordinate dysregulation into individual terms associated with individual state functions, to cast the problem as one of identifying state functions that are *informative* about the phenotype.
- Based on this formulation, we show that the information provided on phenotype by a state function can be bounded from above using statistics of subsets of this subnetwork state. Using this bound, we develop bottom-up enumeration algorithms that can effectively prune out the subnetwork space to identify informative state functions efficiently.
- We use subnetworks identified by the proposed algorithms to train neural networks for classification of phenotype, which are better suited to modeling the combinatorial relationship between the expression levels of genes in a subnetwork, as compared to classifiers that require aggregates of the expression profiles of genes as features (*e.g.*, SVMs).

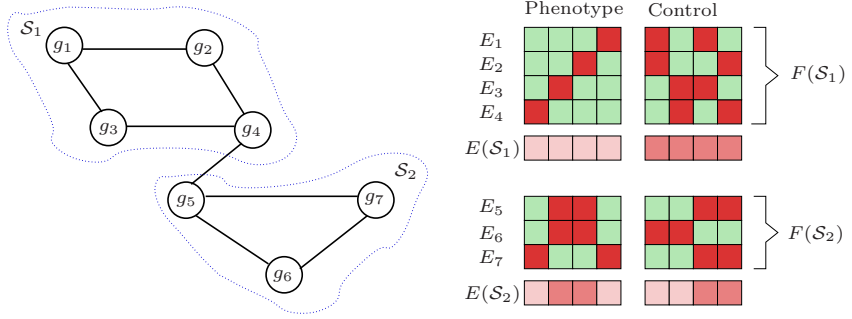
We describe these algorithmic innovations in detail in Section 2.

**Results.** We implement CRANE in Matlab and perform comprehensive cross-classification experiments for prediction of metastasis in CRC. These experiments show that subnetworks identified by the proposed framework significantly outperform subnetworks identified by greedy algorithms in terms of accuracy of classification. We also investigate the highly informative subnetworks in detail to assess their potential in highlighting the mechanisms of metastasis in CRC. We present these results in Section 3 and conclude our discussion in Section 4.

## 2 Methods

In the context of a specific phenotype, a group of genes that exhibit significant differential expression and whose products interact with each other may be useful in understanding the network dynamics of the phenotype. This is because, the patterns of (i) collective differential expression and (ii) connectivity in protein-protein interaction (PPI) network are derived from independent data sources (sample-specific mRNA expression and generic protein-protein interactions, respectively). Thus, they provide corroborating evidence indicating that the corresponding subnetwork of the PPI network may play an important role in the manifestation of phenotype. In this paper, we refer to the collective differential expression of a group of genes as *coordinate dysregulation*. We call a group of coordinately dysregulated genes that induce a connected subnetwork in a PPI network a *coordinately dysregulated subnetwork*.

**Dysregulation of a gene with respect to a phenotype.** For a set  $\mathcal{V}$  of genes and  $\mathcal{U}$  of samples, let  $E_i \in R^{|\mathcal{U}|}$  denote the properly normalized [25] gene expression vector for gene  $g_i \in \mathcal{V}$ , where  $E_i(j)$  denotes the relative expression



**Fig. 1.** Additive *vs.* combinatorial coordinate dysregulation. Genes ( $g$ ) are shown as nodes, interactions between their products are shown as edges. Expression profiles ( $E$ ) of genes are shown by colormaps. Dark red indicates high expression (H), light green indicates low expression (L). None of the genes can differentiate phenotype and control samples individually. Aggregate *subnetwork activity* (average expression) for each subnetwork is shown in the row below its gene expression matrix. The aggregate activity of  $S_1$  can perfectly discriminate phenotype and control, but the aggregate activity of  $S_2$  cannot discriminate at all. For each subnetwork  $S_1$  and  $S_2$ , each column of the gene expression matrix specifies the *subnetwork state* in the corresponding sample. The states of both subnetworks can perfectly discriminate phenotype and control (for  $S_2$ , up-regulation of  $g_7$  alone or  $g_5$  and  $g_6$  together indicates phenotype; we say *state functions* LLH and HHL are indicative of phenotype).

of  $g_i$  in sample  $s_j \in \mathcal{U}$ . Assume that the phenotype vector  $C$  annotates each sample as phenotype or control, such that  $C_j = 1$  indicates that sample  $s_j$  is associated with the phenotype (*e.g.*, taken from metastatic sample) and  $C_j = 0$  indicates that  $s_j$  is a control sample (*e.g.*, taken from a non-metastatic tumor sample). Then, the mutual information  $I(E_i; C) = H(C) - H(C|E_i)$  of  $E_i$  and  $C$  is a measure of the reduction of uncertainty about phenotype  $C$  due to the knowledge of the expression level of gene  $g_i$ . Here,  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$  denotes the Shannon entropy of discrete random variable  $X$  with support  $\mathcal{X}$ . The entropy  $H(E_i)$  of the expression profile of gene  $g_i$  is computed by quantizing  $E_i$  properly. Clearly,  $I(E_i; C)$  provides a reasonable measure of the dysregulation of  $g_i$ , since it quantifies the power of the expression level of  $g_i$  in distinguishing phenotype and control samples.

**Additive coordinate dysregulation.** Now let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  denote a PPI network where the product of each gene  $g_i \in \mathcal{V}$  is represented by a node and each edge  $g_i g_j \in \mathcal{E}$  represents an interaction between the products of  $g_i$  and  $g_j$ . For a subnetwork of  $\mathcal{G}$  with set of nodes  $\mathcal{S} \subseteq \mathcal{V}$ , Chuang *et al.* [20] define the *subnetwork activity* of  $\mathcal{S}$  as  $E_{\mathcal{S}} = \sum_{g_i \in \mathcal{S}} E_i / \sqrt{|\mathcal{S}|}$ , *i.e.*, the aggregate expression profile of the genes in  $\mathcal{S}$ . Then, the dysregulation of  $\mathcal{S}$  is given by  $I(E_{\mathcal{S}}; C)$ , which is a measure of the reduction in uncertainty on phenotype  $C$ , due to knowledge of the aggregate expression level of all genes in  $\mathcal{S}$ . In the following discussion, we refer to  $I(E_{\mathcal{S}}; C)$  as the *additive coordinate dysregulation* of  $\mathcal{S}$ .

**Combinatorial coordinate dysregulation.** Additive coordinate dysregulation is useful for identifying subnetworks that are composed of genes dysregulated in the same direction (either up- or down-regulated). However, interactions among genes and proteins can also be inhibitory (or more complex), and the dysregulation of genes in opposite directions can also be coordinated, as illustrated in Figure 1. Combinatorial formulation of coordinate dysregulation may be able to better capture such complex coordination patterns.

To define combinatorial coordinate dysregulation, we consider binary representation of gene expression data. Binary representation of gene expression is commonly utilized for several reasons, including removal of noise, algorithmic considerations, and tractable biological interpretation of identified patterns. Such approaches are shown to be effective in the context various problems, ranging from genetic network inference [26] to clustering [27] and classification [28]. Ulitsky *et al.* [19] also use binary representation of differential expression to identify dysregulated pathways with respect to a phenotype. There are also many algorithms for effective binarization of gene expression data [29]. For our purposes, let  $\hat{E}_i$  denote the binarized expression profile of gene  $g_i$ . We say that gene  $g_i$  has *high expression* in sample  $s_j$  if  $\hat{E}_i(j) = \mathbf{H}$  and *low expression* if  $\hat{E}_i(j) = \mathbf{L}$ . Then, the *combinatorial coordinate dysregulation* of subnetwork  $\mathcal{S}$  is defined as

$$I(F_{\mathcal{S}}; C) = H(C) - H(C|\hat{E}_1, \hat{E}_2, \dots, \hat{E}_m), \quad (1)$$

where  $F_{\mathcal{S}} = \{\hat{E}_1, \hat{E}_2, \dots, \hat{E}_m\} \in \{\mathbf{L}, \mathbf{H}\}^m$  is the random variable that represents the combination of binary expression states of the genes in  $\mathcal{S}$  and  $m = |\mathcal{S}|$ .

The difference between additive and combinatorial coordinate dysregulation is illustrated in Figure 1. Anastassiou [23] also incorporates this combinatorial formulation to define the synergy between a pair of genes as  $\psi(g_1, g_2) = I(\hat{E}_1, \hat{E}_2; C) - (I(\hat{E}_1; C) + I(\hat{E}_2; C))$ . Generalizing this formulation to the synergy between multiple genes, it can be shown that identification of multiple genes with synergistic dysregulation is an intractable computational problem [23]. Here, we define combinatorial coordinate dysregulation as a more general notion than synergistic dysregulation, in that coordinate dysregulation is defined based solely on collective differential expression, whereas synergy explicitly looks for genes that cannot individually distinguish phenotype and control samples.

Subnetworks that exhibit combinatorial coordinate dysregulation with respect to a phenotype may shed light into the mechanistic bases of that phenotype. However, identification of such subnetworks is intractable, and due to the combinatorial nature of the associated objective function ( $I(F_{\mathcal{S}}; C)$ ), greedy algorithms may not suit well to this problem. This is because, as also demonstrated by the example in Figure 1, it is not straightforward to bound the combinatorial coordinate dysregulation of a subnetwork in terms of the individual dysregulation of its constituent genes or coordinate dysregulation of its smaller subnetworks. Motivated by these considerations, we propose to decompose the combinatorial coordinate dysregulation of a subnetwork into individual subnetwork state functions and show that information provided by state functions of larger subnetworks can be bounded using statistics of their smaller subnetworks.

**Subnetwork state functions informative of phenotype.** Let  $f_S \in \{\text{H}, \text{L}\}^m$  denote an observation of the random variable  $F_S$ , *i.e.*, a specific combination of the expression states of the genes in  $\mathcal{S}$ . By definition of mutual information, we can write the combinatorial coordinate dysregulation of  $\mathcal{S}$  as

$$I(F_S; C) = \sum_{f_S \in \{\text{H}, \text{L}\}^m} J(f_S; C) \quad (2)$$

where

$$J(f_S; C) = p(f_S) \sum_{c \in \{0,1\}} p(c|f_S) \log(p(c|f_S)/p(c)). \quad (3)$$

Here,  $p(x)$  denotes  $P(X = x)$ , that is the probability that random variable  $X$  is equal to  $x$  (similarly,  $p(x|y)$  denotes  $P(X = x|Y = y)$ ). In biological terms,  $J(f_S; C)$  can be considered a measure of the information provided by subnetwork state function  $f_S$  on phenotype  $C$ . Therefore, we say a state function  $f_S$  is *informative* of phenotype if it satisfies the following conditions:

- $J(f_S; C) \geq j^*$ , where  $j^*$  is an adjustable threshold.
- $J(f_S; C) \geq J(f_{\mathcal{R}}; C)$  for all  $f_{\mathcal{R}} \sqsubseteq f_S$ . Here,  $f_{\mathcal{R}} \sqsubseteq f_S$  denotes that  $f_{\mathcal{R}}$  is a substate of state function  $f_S$ , that is  $\mathcal{R} \subseteq \mathcal{S}$  and  $f_{\mathcal{R}}$  maps each gene in  $\mathcal{R}$  to an expression level that is identical to the mapping provided by  $f_S$ .

Here, the first condition ensures that the information provided by the state function is considered high enough with respect to a user-defined threshold. It can be shown that for any  $\mathcal{S} \subseteq \mathcal{V}$ ,  $0 \leq J(f_S; C) \leq \max\{-p(c) \log p(c), -(1-p(c)) \log(1-p(c))\} = j_{\max}(p(c))$  [30], where  $p(c)$  denotes the fraction of phenotype samples among all available samples. Therefore, in practice, we allow the user to specify a threshold  $j^{**}$  in the range  $[0, 1]$  and adjust it as  $j^* = j^{**} j_{\max}(p(c))$ , to make the scoring criterion interpretable and uniform across all datasets. The second condition ensures that informative state functions are non-redundant, that is, a state function is considered informative only if it provides more information on the phenotype than any of its substates can. This restriction ensures that the expression of each gene in the subnetwork provides additional information on the phenotype, capturing the synergy between multiple genes to a certain extent. For a given set of phenotype and control samples and a reference PPI network, the objective of our framework is to identify all informative state functions.

**Algorithms for the identification of informative state functions.** Since the space of state functions is very large, the problem of discovering all informative state functions is intractable. Here, we address this challenge by utilizing a bound on the value of  $J$  to effectively prune the search space. Our approach is inspired by a similar result by Smyth and Goodman [31] on information-theoretic identification of association rules in databases. In the following theorem, we show that the information that can be provided by all superstates of a given state function can be bounded based on the statistics of that state function, without any information about the superstate.

**Theorem 1.** Consider a subnetwork  $\mathcal{S} \subseteq \mathcal{V}$  and associated state function  $f_{\mathcal{S}}$ . For any  $f_{\mathcal{R}} \supseteq f_{\mathcal{S}}$ , the following bound holds:

$$J(f_{\mathcal{R}}; C) \leq p(f_{\mathcal{S}}) \max_{c \in \{0,1\}} \left\{ p(c|f_{\mathcal{S}}) \log \frac{1}{p(c)} \right\} = J_{\text{bound}}(f_{\mathcal{S}}, C). \quad (4)$$

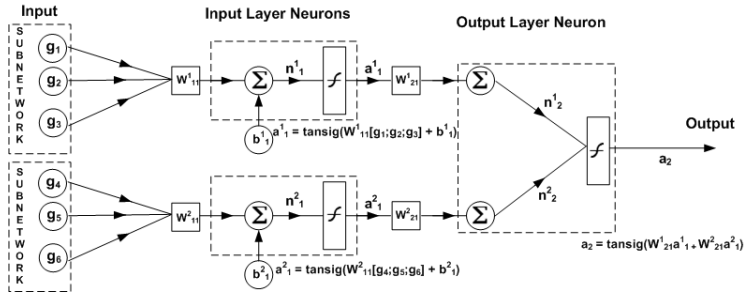
The proof of this theorem is provided in the supplementary materials [30]. Note that this theorem does not state that the  $J$ -value of a state function is bounded by the  $J$ -value of its smaller parts, it rather provides a bound on the  $J$ -value of the larger state function based on simpler statistics of its smaller parts. Using this bound, we develop an algorithm, CRANE, to efficiently search for informative state functions. CRANE enumerates state functions in a bottom-up fashion, by pruning out the search space effectively based on the following principles:

1. A state function  $f_{\mathcal{S}}$  is said to be a *candidate* state function if  $|\mathcal{S}| = 1$  or  $J(f_{\mathcal{S}}; C) \geq J(f_{\mathcal{S} \setminus \{g_i\}}; C)$  for all  $g_i \in \mathcal{S}$ .
2. A candidate state function  $f_{\mathcal{S}}$  is said to be *extensible* if  $J_{\text{bound}}(f_{\mathcal{S}}; C) \geq j^*$ . This restriction enables pruning of larger state functions using statistics of smaller state functions.
3. An extension of state function  $f_{\mathcal{S}}$  is obtained by adding one of the H or L states of a gene  $g_i \in \mathcal{V} \setminus \mathcal{S}$  such that  $g_i g_j \in \mathcal{E}$ , where  $g_j$  is the most recently added gene to  $f_{\mathcal{S}}$ . This ensures network connectivity of the subnetwork associated with the generated state functions.
4. For an extensible state function, all possible extensions are considered and among those that qualify as candidate state functions, the top  $b$  state functions with maximum  $J(\cdot)$  are selected as candidate state functions. Here,  $b$  is an adjustable parameter that determines the breadth of the search and the case  $b = 1$  corresponds to a greedy algorithm.
5. An extensible state function  $f_{\mathcal{S}}$  is not extended if  $|\mathcal{S}| = d$ . Here,  $d$  is an adjustable parameter that determines the depth of the search.

CRANE enumerates all candidate state functions that qualify according to these principles, for given  $j^*$ ,  $b$ , and  $d$ . At the end of the search process, the candidate state functions that are not superceded by another candidate state function (the leaves of the enumeration tree) are identified as informative state functions, if their  $J$ -value exceeds  $j^*$ . A detailed pseudo-code for this procedure is given in the supplementary materials [30].

**Using state functions to predict metastasis in cancer.** An important application of informative state functions is that they can serve as features for classification of phenotype. Since the genes that compose an informative state function are by definition highly discriminative of phenotype and control when considered *together*, they are expected to perform better than single-gene features [20]. Note here that CRANE discovers specific state functions that are informative of phenotype, as opposed to subnetworks that can discriminate phenotype or control. However, by Equation 2, we expect that a high  $J(f_{\mathcal{S}}, C)$  for a specific state function  $f_{\mathcal{S}}$  is associated with a potentially high  $I(F_{\mathcal{S}}, C)$  for the corresponding subnetwork  $\mathcal{S}$ . Therefore, for the application of CRANE in





**Fig. 2.** Neural network model used to utilize subnetworks identified by CRANE for classification. Each subnetwork is represented by an input layer neuron and these neurons are connected to a single output layer neuron.

classification, we sort the subnetworks that are associated with discovered state functions based on their combinatorial coordinate dysregulation  $I(F_S, C)$  and use the top  $K$  disjoint (non-overlapping in terms of their gene content) subnetworks with maximum  $I(F_S, C)$  as features for classification. In the next section, we report results of classification experiments for different values of  $K$ .

Deriving representative features for subnetworks is a challenging task. Using simple aggregates of individual expression levels of genes along with traditional classifiers (*e.g.*, regression or SVMs) might not be adequate, since such representations may not capture the combinatorial relationship between the genes in the subnetwork. For this reason, we use neural networks that incorporate subnetwork states ( $F_S$ ) directly as features. The proposed neural network model is illustrated in Figure 2. In the example of this figure, two subnetworks are used to build the classifier. Each input is the expression level of a gene and the inputs that correspond to a particular subnetwork are connected together to an input layer neuron. All input layer neurons, each representing a subnetwork, are connected to a single output layer neuron, which produces the output. Each layer's weights and biases are initialized with the Nguyen-Widrow layer initialization method (provided by Matlab's `initnw` parameter). Then for a given gene expression dataset for a range of control and phenotype samples (which, in our experiments, is identical to that used for identification of informative state functions), the network is trained with Levenberg-Marquardt backpropagation (using Matlab's `trainlm` parameter), so that, given expression profiles in the training dataset, the output of the second layer matches the associated phenotype vector within minimal mean squared error. This learned model is then used to perform classification tests on a different gene expression dataset for the same phenotype.

### 3 Results and Discussion

In this section, we evaluate the performance of CRANE in identifying state functions associated with metastasis of colorectal cancer (CRC). We first compare the classification performance of the subnetworks associated with these state

functions against single gene markers and subnetworks identified by two greedy algorithms that aim to maximize additive and combinatorial coordinate dysregulation. Then, we inspect the subnetworks that are useful in classification, and discuss the insights these subnetworks can provide into metastasis of CRC.

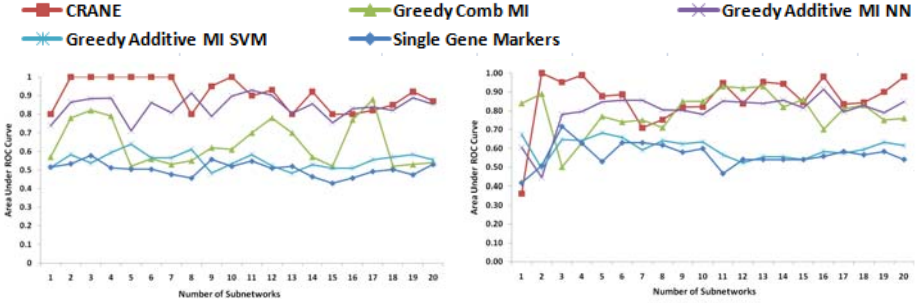
**Datasets.** In our experiments, we use two CRC related microarray datasets obtained from GEO (Gene Expression Omnibus, <http://www.ncbi.nlm.nih.gov/geo/index.cgi>). These datasets, referenced by their accession number in the GEO database, include the following relevant data:

- *GSE6988* contains expression profiles of 17,104 genes across 27 vs. 20 colorectal tumor samples with and without liver metastasis, respectively.
- *GSE3964* contains expression profiles of 5,845 genes across 28 vs. 18 colorectal tumor samples with and without liver metastasis, respectively.

The human protein-protein interaction data used in our experiments is obtained from the Human Protein Reference Database (HPRD), <http://www.hprd.org>. This dataset contains 35023 binary interactions among 9299 proteins, as well as 1060 protein complexes consisting of 2146 proteins. We integrate the binary interactions and protein complexes using a matrix model (e.g., each complex is represented as a clique of the proteins in the complex), to obtain a PPI network composed of 42781 binary interactions among 9442 proteins.

**Experimental design.** For each of the datasets mentioned above, we discover informative state functions (in terms of discriminating tumor samples with or without metastasis) using CRANE. While state functions that are indicative of either metastatic or non-metastatic phenotype can have high  $J(\cdot)$  values, we use only those that are indicative of (*i.e.*, knowledge of which increases the likelihood of) metastatic phenotype for classification and further analyses, since such state functions are directly interpretable in terms of their association with metastasis. In the experiments reported here, we set  $b = 10$ , and  $d = 6$ . The value of  $j^{**}$  is set to 0.5, that is subnetwork state functions that have at least as half as the maximum achievable  $J$ -value for the given dataset are considered informative. Note that these parameters are used to balance the trade-off between computational cost of subnetwork identification and classification accuracy. The reported values are those that provide reasonable performance by spending a reasonable amount of time on subnetwork identification (a few hours in Matlab for each dataset). To binarize the gene expression datasets, we first normalize the gene expression profiles so that each gene has an average expression of 0 and standard deviation 1. Then we set the top  $\alpha$  fraction of the entries in the normalized gene expression matrix to H (high expression) and the rest to L (low expression). In the reported experiments, we use  $\alpha = 0.25$  (25% of the genes are expressed on an average) as this value is found to optimize the classification performance.

**Implementation of other algorithms.** We also use two greedy algorithms to identify coordinately dysregulated subnetworks, one of which aims to maximize additive coordinate dysregulation [20], while the other aims to maximize combinatorial coordinate dysregulation. We implement the greedy algorithms to



Training: GSE6988, Testing: GSE3964

Training: GSE3964, Testing: GSE6988

**Fig. 3.** Classification performance of subnetworks identified by CRANE in predicting colon cancer metastasis, as compared to those identified by greedy algorithms that aim to maximize combinatorial or additive coordinate dysregulation, as well as single-gene markers. Subnetworks identified by CRANE and greedy combinatorial algorithm are used to train neural networks (NNs), while those identified by the greedy additive algorithm are used to train NNs, as well as support vector machines (SVMs). In the graphs, horizontal axes show the number of disjoint subnetwork features (with maximum combinatorial or greedy coordinate dysregulation) used in classification, vertical axes show the area under ROC curve achieved by the corresponding classifier.

identify a subnetwork associated with each gene in the network by seeding the greedy search process from that gene. The greedy algorithms grow subnetworks by iteratively adding to the subnetwork a network neighbor of the genes that are already in the subnetwork. At each iteration, the neighbor that maximizes the coordinate dysregulation of the subnetwork is selected to be added. Once all subnetworks are identified, we sort these subnetworks according to their coordinate dysregulation ( $I(E_S; C)$  or  $I(F_S; C)$ ) and use the top  $K$  disjoint subnetworks to train and test classifiers, for different values of  $K$ . The binarization scheme for greedy identification of combinatorially dsregulated subnetworks is identical to that for CRANE. While quantizing  $E_S$  to compute  $I(E_S; C)$ , as suggested in [20], we use  $\lfloor \log_2(|\mathcal{U}|) \rfloor + 1$  bins where  $|\mathcal{U}|$  denotes the number of samples. Note that, in [20], the subnetworks identified by the greedy algorithm are filtered through three statistical tests. In our experiments, these statistical tests are not performed for the subnetworks discovered by any of the three algorithms.

The design of classifiers for combinatorially dsregulated subnetworks identified by the greedy algorithm is also identical to that for subnetworks identified by CRANE. For the subnetworks with additive coordinate dysregulation, we compute the subnetwork activity  $E_S$  for each subnetwork, and use these as features to train and test two different classifiers: (i) a support vector machine (SVM) using Matlab’s `svmtrain` and `svmclassify` functions (this method is not applicable to combinatorial coordinate dysregulation), (ii) feed-forward neural networks, in which each input represents the subnetwork activity for a subnetwork and these inputs are connected to hidden layer neurons. For the single-gene markers, we rank all genes according to the mutual information of their expression

profile with phenotype ( $I(E_i; C)$ ) and use the expression level of  $K$  genes with maximum  $I(E_i; C)$  as features for classification.

**Classification performance.** We evaluate the cross-classification performance of the subnetworks in the context of predicting metastasis of CRC. Namely, we use subnetworks discovered on the *GSE6988* dataset to train classifiers and we test the resulting classifiers on *GSE3964*. Similarly, we use subnetworks discovered on *GSE3964* to train classifiers using the same dataset and perform testing of these classifiers on *GSE6988*. The cross-classification performance of subnetworks discovered by an algorithm is not only indicative of the power of the algorithm in discovering subnetworks that are descriptive of phenotype, but also the reproducibility of these subnetworks across different datasets.

The classification performance of the subnetworks identified by CRANE and greedy algorithms is compared in Figure 3. In the figure, for each  $1 \leq K \leq 20$ , the ‘Area Under ROC Curve’(AUC) is reported for each classifier. AUC is a measure of the overall performance of a classifier, which accounts for the trade-off between the precision (selectivity) and recall (sensitivity) of predictions. Here, precision is defined as the fraction of true positives among all samples classified as phenotype by the classifier, while recall is defined as the fraction of true positives among all true phenotype samples. AUC is a measure of the average precision across varying values of recall and an AUC of 1.0 indicates that the classifier provides perfect precision without sacrificing recall (or vice versa).

As seen in Figure 3, subnetworks identified by CRANE significantly outperform the subnetworks identified by other algorithms in predicting metastasis of colorectal cancer. In fact, in both cases, CRANE has the potential to deliver 100% accuracy using very few subnetworks. While we use a simple feature selection method here for purposes of illustration, the performance of CRANE subnetworks are quite consistent, suggesting that these performance figures can indeed be achieved by developing elegant methods for selection of subnetwork features. These results are rather impressive, given that the best performance that can be achieved by the greedy additive algorithm is 91% and 93% for the classification of *GSE3964* and *GSE6988*, respectively. On the other hand, the greedy algorithm for combinatorial coordinate dysregulation is outperformed by the greedy additive algorithm on the classification of *GSE3964* and performs quite poorly compared to CRANE. These results show that, besides the combinatorial formulation of coordinate dysregulation, the search algorithm implemented by CRANE

**Table 1.** Five subnetworks that are associated with the most informative state functions discovered on GSE6988

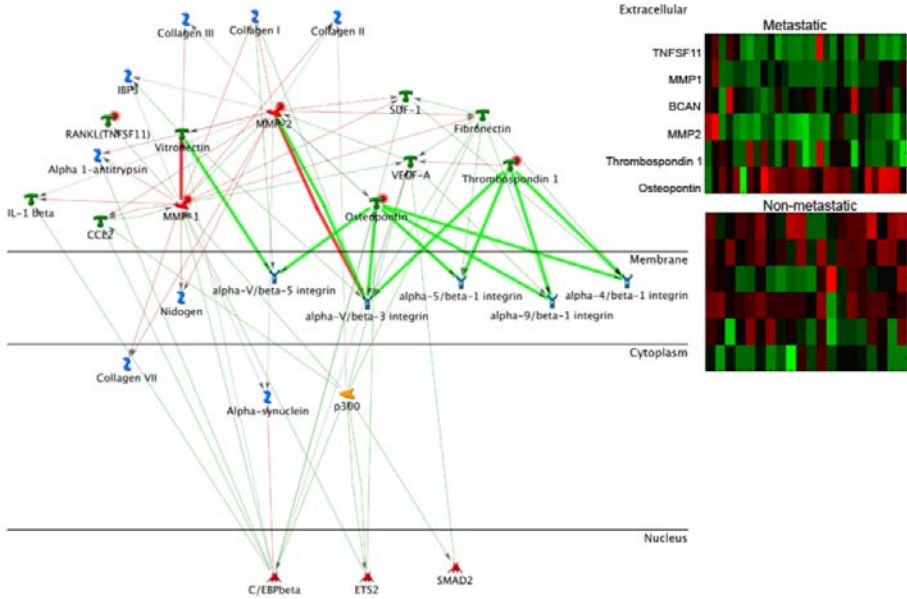
Rank	Proteins	Comb. Coord. Dysregulation	Most Significantly Enriched Process	Enrichment p-value
1	CASP1, LMNA, CTCF, APP, APBA1	1.00	Cell Adhesion	$1 \times 10^{-6}$
2	JAK2, STAT5A, IL7R, STAT3, IL2RA	0.96	Lymphocyte Proliferation	$1 \times 10^{-9}$
3	TRAF1, CFLAR, NFKB1, FBXW11, NFKBIB	0.96	Inflammation	$1 \times 10^{-8}$
4	CD9, KIT, BTK, WAS, NCK1	0.96	Cell Adhesion	$1 \times 10^{-4}$
5	XRCC5, VAV1, ARGHDIA, RAC2, NOS2A	0.96	Inflammation	$1 \times 10^{-4}$

also adds to the power of identified subnetworks in discriminating metastatic and non-metastatic samples.

**Effect of parameters.** We also investigate the effect of parameters used to configure CRANE on classification performance, by fixing all but one of the parameters to the above-mentioned values and varying the remaining parameter. The results of these experiments are given in detail in supplementary materials [30]. To summarize, we observe that classification performance is quite robust against variation in  $\alpha$  ranging from 10% to 40%. As expected, classification performance improves by increasing  $j^{**}$ , but values of  $j^{**}$  as low as 0.15 still provide nearly 65% average classification accuracy. While increasing  $d$  improves performances as would be expected, this improvement saturates for  $d > 3$  and performance declines for larger subnetworks. This observation can be attributed to curse of dimensionality, since the number of possible values of random variable  $F$  grows exponentially with increasing subnetwork size. Finally, while larger  $b$  improves classification performance in general by increasing the breadth of the search, we observe some exceptions to this behavior (*e.g.*, the average performance for  $b = 3$  appears to be higher than that for  $b = 5$ ).

**Subnetworks and state functions indicative of metastasis in CRC.** Cancer metastasis involves the rapid proliferation and invasion of malignant cells into the bloodstream or lymphatic system. The process is driven, in part, by the dysregulation of proteins involved in cell adhesion and motility [32], the degradation of the extracellular matrix (ECM) at the invasive front of the primary tumor [33], and is associated with chronic inflammation [34]. An enrichment analysis of the top five subnetworks identified on *GSE6988* reveals that all of these subnetworks are highly significant for the network processes underlying these phenotypes (Table 1).

Further, as CRC metastasis is our classification endpoint, we wanted to evaluate our subnetworks in terms of their potential to propose testable hypotheses. In particular, to highlight the power of our model approach, we choose a subnetwork for which at least one gene was expressed in the state function indicative of CRC metastasis. This subnetwork contains TNFSF11, MMP1, BCAN, MMP2, TBSH1, and SPP1 and the state function LLLLLH (in respective order) indicates metastatic phenotype with  $J$ -value 0.33. The combinatorial dysregulation of this subnetwork is 0.72, while its additive coordinate dysregulation is 0.37, *i.e.*, this is a subnetwork which would likely have escaped detection by the greedy method based on additive dysregulation (this subnetwork is not listed in Table 1 since it is not among the top five scoring subnetworks). Using the genes in this subnetwork as a seed, we construct a small subnetwork diagram for the purpose of more closely analyzing the post-translational interactions involving these proteins. This is done using Metacore, a commercial platform that provides curated, highly reliable interactions. From this subnetwork, we remove all genes indicated to be not expressed in human colon by the database, and then selectively prune it in order to clearly focus on a particular set of interactions (Figure 4). It merits noting that although Brevican (BCAN) is in subnetwork, it is removed for being



**Fig. 4.** Hypothesis-driver subnetwork - interaction diagram illustrating key interactions with gene products from a subnetwork identified by CRANE as indicative of CRC metastasis. Shown are the gene products in discovered subnetwork (red circles) and their direct interactions with other proteins. Green lines represent an activating interaction, red lines indicate an inhibitory interaction. Arrows indicate direction of mRNA. Inset is the expression pattern of subnetwork proteins at the level of mRNA.

non-expressed in the human colon, although evidence from the Gene Expression Omnibus (see accession *GDS2609*) casts doubt on this, as does the microarray we use for scoring (*GSE6988*).

As seen on the interaction diagram, SPP1 (Osteopontin) and TBSH1 (Thrombospondin 1) interact with a number of the integrin heterodimers to increase their activity (green line). Integrin heterodimers play a major role in mediating cell adhesion and cell motility. SPP1, up-regulated in metastasis (see inset in Figure 4), is a well-studied protein that triggers intracellular signaling cascades upon binding with various integrin heterodimers, promotes cell migration when it binds CD44, and when binding the alpha-5/beta-3 dimer in particular, promotes angiogenesis, which is associated with the metastatic phenotype of many cancers [35]. MMP proteins are involved in the breakdown of ECM, particularly collagen which is the primary substrate at the invasive edge of colorectal tumors [36]. MMP-1 has an inhibitory effect on Vitronectin (red line), hence the loss of expression of MMP-1 may “release the brake” on Vitronectin, which in turn may increase the activity of the alpha-v/beta-5 integrin heterodimer. Likewise, MMP-2 shows an inhibitory interaction with the alpha-5/beta-3 dimer, which may counteract to some extent the activating potential of SPP1,

suggesting that a loss of MMP-2 may exacerbate the metastatic phenotype. Taken together, these interactions suggest a number of perturbation experiments, perhaps by pharmacological inhibition or siRNA interference of the integrin dimmers or MMP proteins, to evaluate the role of these interactions, individually or synergistically, in maintaining the metastatic phenotype. Note also that, alpha-v/beta-5 integrin does not exhibit significant differential expression at the mRNA-level, suggesting that the state function identified by CRANE may be a signature of its post-translational dysregulation in metastatic cells.

## 4 Conclusion

We present a novel framework for network based analysis of coordinate dysregulation in complex phenotypes. Experimental results on metastasis of colorectal cancer show that the proposed framework can achieve almost perfect performance when discovered subnetworks are used as features for classification. These results are highly promising in that the state functions that are found to be informative of metastasis can also be useful in modeling the mechanisms of metastasis in cancer. Detailed investigation of the state functions and the interactions between proteins that together compose state functions might therefore lead to development of novel hypotheses, which in turn may be useful for development of therapeutic intervention strategies for late stages of cancer.

## Acknowledgments

This work is supported, in part, by NSF CAREER Award CCF-0953195 and NIH Grant, UL1-RR024989 Supplement, from the National Center for Research Resources (Clinical and Translational Science Awards).

## References

1. Schadt, E.E., Lamb, J., Yang, X., Zhu, J., Edwards, S., Guhathakurta, D., Sieberts, S.K., Monks, S., Reitman, M., Zhang, C., Lum, P.Y., Leonardson, A., Thieringer, R., Metzger, J.M., Yang, L., Castle, J., Zhu, H., Kash, S.F., Drake, T.A., Sachs, A., Lusk, A.J.: An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics* 37(7), 710–717 (2005)
2. Papin, J.A., Hunter, T., Palsson, B.O., Subramaniam, S.: Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology* 6(2), 99–111 (2005)
3. Ideker, T., Sharan, R.: Protein networks in disease. *Genome Res.* 18(4), 644–652 (2008)
4. Rich, J., Jones, B., Hans, C., Iversen, E., McClendon, R., Rasheed, A., Bigner, D., Dobra, A., Dressman, H., Nevins, J., West, M.: Gene expression profiling and genetic markers in glioblastoma survival. *Cancer Research* 65, 4051–4058 (2005)



5. Ewing, R.M., Chu, P., Elisma, F., Li, H., Taylor, P., Climie, S., McBroom-Cerajewski, L., Robinson, M.D., O'Connor, L., Li, M., Taylor, R., Dharsee, M., Ho, Y., Heilbut, A., Moore, L., Zhang, S., Ornatsky, O., Bukhman, Y.V., Ethier, M., Sheng, Y., Vasilescu, J., Abu-Farha, M., Lambert, J.P.P., Duewel, H.S., Stewart, I.I., Kuehl, B., Hogue, K., Colwill, K., Gladwish, K., Muskat, B., Kinach, R., Adams, S.L.L., Moran, M.F., Morin, G.B., Topaloglou, T., Figeys, D.: Large-scale mapping of human protein-protein interactions by mass spectrometry. *Molecular systems biology* 3 (2007)
6. Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabasi, A.L.: The human disease network. *PNAS* 104(21), 8685–8690 (2007)
7. Rhodes, D.R., Chinnaiyan, A.M.: Integrative analysis of the cancer transcriptome. *Nat. Genet.* 37(suppl.) (June 2005)
8. Franke, L., Bakel, H., Fokkens, L., de Jong, E.D., Egmont-Petersen, M., Wijmenga, C.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.* 78(6), 1011–1025 (2006)
9. Karni, S., Soreq, H., Sharan, R.: A network-based method for predicting disease-causing genes. *Journal of Computational Biology* 16(2), 181–189 (2009)
10. Lage, K., Karlberg, O.E., Størling, Z.M., Páll, P.A.G., Rigina, O., Hinsby, A.M., Tümer, Z., Pociot, F., Tommerup, N., Moreau, Y., Brunak, S.: A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature Biotechnology* 25(3), 309–316 (2007)
11. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering regulatory and signalling circuits in molecular interaction networks. In: *ISMB*, pp. 233–240 (2002)
12. Guo, Z., Li, Y., Gong, X., Yao, C., Ma, W., Wang, D., Li, Y., Zhu, J., Zhang, M., Yang, D., Wang, J.: Edge-based scoring and searching method for identifying condition-responsive protein–protein interaction sub-network. *Bioinformatics* 23(16), 2121–2128 (2007)
13. Nacu, Ș., Critchley-Thorne, R., Lee, P., Holmes, S.: Gene expression network analysis and applications to immunology. *Bioinformatics* 23(7), 850–858 (2007)
14. Liu, M., Liberzon, A., Kong, S.W., Lai, W.R., Park, P.J., Kohane, I.S., Kasif, S.: Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics* 3(6), e96 (2007)
15. Cabusora, L., Sutton, E., Fulmer, A., Forst, C.V.: Differential network expression during drug and stress response. *Bioinformatics* 21(12), 2898–2905 (2005)
16. Patil, K.R., Nielsen, J.: Uncovering transcriptional regulation of metabolism by using metabolic network topology. *PNAS* 102(8), 2685–2689 (2005)
17. Scott, M.S., Perkins, T., Bunnell, S., Pepin, F., Thomas, D.Y., Hallett, M.: Identifying regulatory subnetworks for a set of genes. *Mol. Cell Prot.*, 683–692 (2005)
18. Chowdhury, S.A., Koyutürk, M.: Identification of coordinately dysregulated sub-networks in complex phenotypes. In: *PSB*, pp. 133–144 (2010)
19. Ulitsky, I., Karp, R.M., Shamir, R.: Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In: Vingron, M., Wong, L. (eds.) *RECOMB 2008. LNCS (LNBI)*, vol. 4955, pp. 347–359. Springer, Heidelberg (2008)
20. Chuang, H.Y., Lee, E., Liu, Y.T., Lee, D., Ideker, T.: Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* 3 (October 2007)
21. Nibbe, R.K., Ewing, R., Myeroff, L., Markowitz, M., Chance, M.: Discovery and scoring of protein interaction sub-networks discriminative of late stage human colon cancer. *Mol. Cell Prot.* 9(4), 827–845 (2009)
22. Nibbe, R.K., Koyutürk, M., Chance, M.R.: An integrative -omics approach to identify functional sub-networks in human colorectal cancer. *PLoS Comput. Biol.* 6(1), e1000639 (2010)



23. Anastassiou, D.: Computational analysis of the synergy among multiple interacting genes. *Mol. Syst. Biol.* 3(83) (2007)
24. Watkinson, J., Wang, X., Zheng, T., Anastassiou, D.: Identification of gene interactions associated with disease from gene expression data using synergy networks. *BMC Systems Biology* 2(1) (2008)
25. Quackenbush, J.: Microarray data normalization and transformation. *Nat. Genet.* 32(suppl.), 496–501 (2002)
26. Akutsu, T., Miyano, S., Kuhara, S.: Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In: *Pacific Symposium on Biocomputing*, pp. 17–28 (1999)
27. Koyutürk, M., Szpankowski, W., Grama, A.: Biclustering gene-feature matrices for statistically significant dense patterns. In: *IEEE Computational Systems Bioinformatics Conference (CSB 2004)*, pp. 480–484 (2004)
28. Akutsu, T., Miyano, S.: Selecting informative genes for cancer classification using gene expression data. In: *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, pp. 3–6 (2001)
29. Shmulevich, I., Zhang, W.: Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics* 18(4), 555–565 (2002)
30. Chowdhury, S.A., Nibbe, R.K., Chance, M.R., Koyutürk, M.: Supplement to “Subnetwork state functions define dysregulated subnetworks in cancer”, [http://vorlon.case.edu/~mxk331/crane/recomb2010\\_supplement.pdf](http://vorlon.case.edu/~mxk331/crane/recomb2010_supplement.pdf)
31. Smyth, P., Goodman, R.M.: An information theoretic approach to rule induction from databases. *IEEE Trans. on Knowl. and Data Eng.* 4(4), 301–316 (1992)
32. Paschos, K., Canovas, D., Bird, N.: The role of cell adhesion molecules in the progression of colorectal cancer and the development of liver metastasis. *Cell Signal* 21(5), 665–674 (2009)
33. Zucker, S., Vacirca, J.: Role of matrix metalloproteinases (mmps) in colorectal cancer. *Cancer Metastasis Rev.* 23(1-2), 101–117 (2004)
34. McConnell, B., Yang, V.: The role of inflammation in the pathogenesis of colorectal cancer. *Curr. Colorectal Cancer Rep.* 5(2), 69–74 (2009)
35. Markowitz, S., Bertagnolli, M.: Molecular origins of cancer: Molecular basis of colorectal cancer. *N. Engl. J. Med.* 361(25), 2449–2460 (2009)
36. Vishnubhotla, R., Sun, S., Huq, J., Bulic, M., Ramesh, A.: Rock-ii mediates colon cancer invasion via regulation of mmp-2 and mmp-13 at the site of invadopodia as revealed by multiphoton imaging. *Laboratory Investigation* 87, 1149–1158 (2007)

# Proteome Coverage Prediction for Integrated Proteomics Datasets

Manfred Claassen<sup>1,2,3</sup>, Ruedi Aebersold<sup>2</sup>, and Joachim M. Buhmann<sup>1</sup>

<sup>1</sup> Department of Computer Science, ETH Zurich

<sup>2</sup> Institute of Molecular Systems Biology, ETH Zurich

<sup>3</sup> Center for Systems Physiology and Metabolic Diseases, Zurich

manfredc@inf.ethz.ch, aebersold@imsb.biol.ethz.ch, jbuhmann@inf.ethz.ch

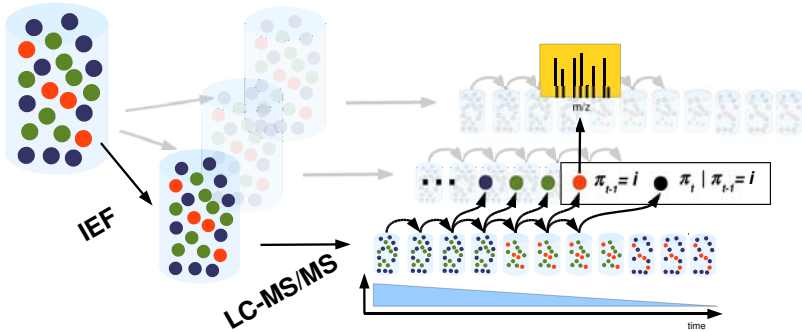
**Abstract.** Comprehensive characterization of a proteome defines a fundamental goal in proteomics. In order to maximize proteome coverage for a complex protein mixture, i.e. to identify as many proteins as possible, various different fractionation experiments are typically performed and the individual fractions are subjected to mass spectrometric analysis. The resulting data are integrated into large and heterogeneous datasets. Proteome coverage prediction refers to the task of extrapolating the number of protein discoveries by future measurements conditioned on a sequence of already performed measurements. Proteome coverage prediction at an early stage enables experimentalists to design and plan efficient proteomics studies. To date, there does not exist any method that reliably predicts proteome coverage from integrated datasets. We present a generalized hierarchical Pitman-Yor process model that explicitly captures the redundancy within integrated datasets. We assess the proteome coverage prediction accuracy of our approach applied to an integrated proteomics dataset for the bacterium *L. interrogans* and we demonstrate that it outperforms ad hoc extrapolation methods and prediction methods designed for non-integrated datasets. Furthermore, we estimate the maximally achievable proteome coverage for the experimental setup underlying the *L. interrogans* dataset. We discuss the implications of our results to determine rational stop criteria and their influence on the design of efficient and reliable proteomics studies.

## 1 Introduction

Recent developments in mass spectrometry based proteomics have enabled biologists to comprehensively characterize proteomes, the protein inventories of biological samples [1]. To achieve extensive proteome coverage, a range of different experiments have to be carefully planned and extensively repeated. Proteome coverage prediction denotes the task of estimating the expected yield of protein discoveries upon experiment repetitions. This task is essential to guide experimental planning and to infer maximal coverage for a particular series of experiments. Here we present a generalized hierarchical Pitman-Yor process to reliably predict proteome coverage for multidimensional fractionation experiments.

The most successful strategy to achieve extensive proteome coverage is referred to as shotgun proteomics. Briefly, proteins are biochemically extracted from a biological sample and are enzymatically digested to yield a complex ensemble of peptides. Protein and/or peptide ensembles are optionally further fractionated according to physical/chemical/biological properties (multidimensional fractionation). Tandem mass spectrometry is then used to sample and identify individual peptide species present in the resulting ensembles and to finally recover the set of proteins initially present in the biological sample [2] (Fig. 1).

The capacity of mass spectrometers limits the number of peptides possibly identified at a time. Due to this constraint it is by far too difficult to identify the entirety of species in a peptide ensemble arising after enzymatic digestion of a typical complex biological sample such as a complete proteome. Two experimental routes are pursued to circumvent this limitation and to enable comprehensive characterization of a complex peptide ensemble. First, peptide ensembles are fractionated into a multitude of less complex and, therefore, more tractable ensembles before being analyzed by tandem mass spectrometry and second, experiments are extensively repeated. Popular fractionation schemes separate peptides with respect to properties such as e.g. size or isoelectric point. Reversed phase liquid chromatography (LC) is the most common fractionation technique



**Fig. 1.** Illustration of a typical multidimensional fractionation experiment. The initial root peptide ensemble obtained from the biological source is separated by some fractionation method (e.g. isoelectric focussing (IEF)), giving rise to a set of related peptide ensembles. LC-MS/MS analysis is performed for each of these fractions. Liquid chromatography fractionation generates a sequence of child peptide ensembles from the root ensemble. Each of these ensembles is derived from the root ensemble by pooling peptides of similar polarity. The sequence of ensembles features descending overall polarity in the course of the experiment. During the experiment peptides  $\pi_t$  are drawn from the sequence of ensembles and analyzed by the mass spectrometer coupled to the liquid chromatography system and subsequently identified computationally. We propose a non-parametric Bayesian approach to characterize the distributions governing the peptide ensembles. We simulate further experiments and thereby predict proteome coverage by sampling from these peptide distributions. For details please refer to section [2].

and separates peptide ensembles according to hydrophobicity and is typically directly coupled to a tandem mass spectrometry system (LC-MS/MS). Multidimensional fractionation strategies comprise multiple steps of fractionation, typically fractionation according to some physico-chemical property other than hydrophobicity followed by LC-MS/MS analysis. (Fig. 1). Shotgun proteomics studies that achieved significant proteome coverage for a variety of organisms have shown to build on extensive repetition of multidimensional fractionation experiments (see e.g. 3).

Methods for proteome coverage prediction estimate the expected number of peptide/protein discoveries when experiments are repeated. Proteome coverage prediction is essential for rational experimental planning of shotgun proteomics studies. Projects aiming at extensive proteome coverage require a considerable amount of experimentation. Proteome coverage should ideally increase efficiently with consecutive experiments. The choice between competing experimental setups should thus be guided by their potential to increase proteome coverage. Methods for proteome coverage prediction enable to rationally determine the optimal setup. Proteome coverage prediction furthermore enables to estimate the maximal coverage as well the volume of experiments required to achieve this coverage.

Proteome coverage prediction and related tasks have not been addressed until recently. Fenyo *et al.* conducted simulation studies to generally investigate how fractionation of peptide or protein ensembles might affect the efficiency of shotgun proteomics experiments 4. Brunner *et al.* roughly estimated upper and lower bounds for proteome coverage from a real data set by assuming worst/best case scenarios 3. Recently, an infinite Markov model based on Dirichlet processes 5 has been proposed to characterize LC-MS/MS experiments and for the first time to predict proteome coverage for one dimensional fractionation experiments 6.

In practice, it is highly desirable to predict proteome coverage of multidimensional fractionation experiments since these strategies have shown to have the largest potential to map out a proteome. However, there does not exist any method for proteome coverage prediction of these experiments. This task is particularly challenging since the proteomes represented by each fraction overlap to an unknown extent. Proteome coverage prediction methods for multidimensional fractionation experiments have to account for this phenomenon.

In this paper we generalize the non-parametric approach to characterize peptide distributions arising in LC-MS/MS experiments 6 to further enable proteome coverage prediction from integrated datasets compiled from multidimensional fractionation experiments. Specifically, we propose a novel generalized hierarchical Pitman-Yor process 7,8 with self-referential base measures that addresses the issue of distribution overlap which is introduced by the fractionation preceding the LC-MS/MS analysis. Besides the possibility to characterize peptide distributions arising in the course of multidimensional fractionation experiments, this approach also lends itself to characterize the biologically more relevant protein distributions. We assess our method on a set of 24 experiments

from multidimensional fractionation of a *L. interrogans* whole proteome sample and report better performance than ad hoc extrapolation schemes and other approaches designed for one dimensional fractionation experiments. We discuss our results with respect to maximally achievable proteome coverage from a peptide- as well as protein-centric perspective.

## 2 Methods

The following sections give technical background and details on the hierarchical Pitman-Yor process framework for proteome coverage prediction based on integrated datasets. Briefly, our approach characterizes the peptide/protein distributions arising in a multidimensional fractionation experiment and simulates further experiments by sampling from these distributions. Proteome coverage is predicted by counting the number of novel peptide/protein discoveries in the simulations. In the following sections we will assume a peptide-centric view for clarity, i.e. consider peptide distributions instead of its protein counterparts. Note that peptides, by virtue of being protein fragments, also refer to protein identities. Therefore, the following sections can also be read by consequently substituting peptides with proteins. Complications arising from peptides ambiguously referring to several protein identities are discussed in section 4.

### 2.1 Pitman-Yor Processes

We apply Pitman-Yor Processes to characterize peptide distributions arising in the course of a series of proteomics experiments. In the following we briefly review the concept of Pitman-Yor Processes in the context of this work.

Like the Gaussian distribution is an appropriate distribution for a real valued random variable in numerous applications, the Pitman-Yor process frequently is an appropriate distribution for complex objects such as discrete distributions [9]. Loosely spoken, Pitman-Yor processes are suited as priors over discrete distributions that are expected to have most of their probability mass on a small number of atoms and only little probability mass on the vast majority of atoms [8]. As various proteomics studies have shown that protein/peptide frequencies exhibit such a property (see e.g. [10]), we use Pitman-Yor processes as priors for distributions  $G$  over a set  $\Pi$  of peptides defined by a protein database of the studied organism.

$$G \mid \gamma, d, H \sim \text{PY}(\gamma, d, H) \quad (1)$$

where  $\text{PY}(\gamma, d, H)$  is a Pitman-Yor process with a concentration parameter  $\gamma$ , a discount parameter  $d$  and a base probability measure  $H$ . The base measure is defined over  $\Pi$  (sample space).  $H$  is frequently chosen uniform, assigning  $1/|\Pi|$  probability mass to each  $\pi \in \Pi$ .

The so called *Chinese Restaurant* construction [11][12] provides an intuitive way to see which kind of distributions are likely to be drawn from a Pitman-Yor process  $\text{PY}(\gamma, d, H)$ . Imagine a restaurant with an infinite number of tables. At

each table a specific dish is served. We construct a distribution  $G$  over dishes after having seated an infinite number of customers. Customers are seated according to a probabilistic rule. Specifically, the probability of the  $t$ -th customer being seated at the table serving dish  $\pi_t = k$  assumes the values

$$P(\pi_t = k \mid \pi_1, \dots, \pi_{t-1}, \gamma, d, H) = \begin{cases} \frac{n_k - d}{t-1+\gamma} & \text{populated table} \\ \frac{\gamma + kd}{t-1+\gamma} & \text{next unpopulated table} \end{cases} \quad (2)$$

where  $n_k$  corresponds to the number of customers already sitting at the table serving dish  $i$ . In case a customer happens to be seated at a new table, the dish served at this table is drawn from the base probability measure  $H$ . A procedural description of serving a new customer in a restaurant with seating arrangement  $R = n_1, n_2, \dots$  is as follows:

```
SEAT( $R, \gamma, d, H$ )
1   $t \leftarrow \text{SAMPLETABLE}(R, \gamma, d)$ 
2  if  $t \neq \text{new}$ 
3    then return  $\text{DISH}(R, t)$ 
4    else return  $\text{SAMPLE}(H)$ 
```

The larger the concentration parameter  $\gamma$ , the higher the chances that a new customer is seated at a new table. The more customers have already been seated, the less likely a new dish will be served. The larger the discount parameter  $d$  the less likely a customer is seated at an already populated table. Note that  $d < 1$ . In summary, the parameters  $\gamma$  and  $d$  control, though in different ways, the deviation of  $G$  from the base measure  $H$ . The *Chinese Restaurant* construction specifies the posterior to iteratively sample from  $\pi_t \mid \pi_1, \dots, \pi_{t-1}, \gamma, d, H$  after marginalizing out  $G$ .

Pitman-Yor Processes are generalizations of the more commonly known Dirichlet processes [11][13]. More precisely, a Dirichlet Process  $\text{DP}(\gamma, H)$  is equivalent to a Pitman-Yor process  $\text{PY}(\gamma, d, H)$  with  $d = 0$ . Both Dirichlet and Pitman-Yor processes will be used as priors for peptide distributions that arise in the course of a multidimensional fractionation experiment. After having estimated the process parameters we will simulate further experiments by sampling according to the *Chinese Restaurant* construction.

## 2.2 Hierarchical Process Model for Multidimensional Fractionation Experiments

In the following we characterize the distributions which arise in a multidimensional fractionation experiment. We specifically describe a typical setup that comprises two consecutive fractionation steps, where the first step splits the initial peptide ensemble into a set of  $I$  fractions that are each analyzed by LC-MS/MS (Fig. 1). Besides enforcing consistency along subsequent fractionation steps using hierarchical processes, we further want our model to explicitly capture the similarity of corresponding peptide distributions across different fractions.

The initial peptide ensemble follows the root distribution  $G$ . We assume a Pitman-Yor process prior  $\text{PY}(\gamma_r, d_r, H)$  for  $G$ . The base measure  $H$  is chosen to be the uniform distribution over the peptides defined by the protein database of the studied organism.

Peptides are not directly sampled from the root distribution  $G$ . Consider some time point  $t$  during the LC-MS/MS analysis of fraction  $i$ . The peptide  $\pi_t^i$  is sampled from the child peptide distribution  $G_t^i$  of the peptide ensemble currently eluting from the liquid chromatography column. Following [6] we assume that the precedent peptide  $\pi_{t-1}^i := j$  is indicative for the current polarity of the chromatography and thereby the current peptide distribution, i.e. with a slight abuse of notation we assume  $G_t^i = G_j^i$ . Further we assume a Dirichlet process prior for  $G_j^i$ , resulting in an infinite Markov model for LC-MS/MS experiments similar to [6].

$$\begin{aligned} G_j^i \mid \gamma_c^i, A_j^i &\sim \text{DP}(\gamma_c^i, A_j^i) \\ \pi_t \mid \pi_{t-1}^i = j &\sim G_j^i \end{aligned} \quad (3)$$

We want the child distributions  $G_j^i$  to be consistent with the root distribution  $G$ , i.e. we want to ensure that peptides having zero probability mass in the initial peptide ensemble still have zero probability mass during an LC-MS/MS experiment. This notion is captured by choosing  $G$  as base measure  $A_j^i$  in (3), yielding a hierarchical process [7]. This choice ensures (1) that  $G_j^i$  is consistent with  $G$ , i.e. the support of  $G_j^i$  is enclosed by the support of  $G$  and (2) that  $G_j^i$  will have similarity to  $G$  to an extent defined by the concentration parameter  $\gamma_c^i$ . Furthermore, we want to capture the similarity between  $G_j^i$  and its corresponding distributions  $G_j^{i'}$  in all other fractions  $i' \neq i$ . Therefore we extend the base measure  $A_j^i$  in (3) to a (self-referential) linear combination of the distributions  $(G_j^{i'})_{i'=1}^I$  and  $G$ .

$$A_j^i = a_i^i G + \sum_{i' \neq i} a_{i'}^i G_j^{i'} \quad (4)$$

Since the values  $a^i := (a_{i'}^i)_{i'=1}^I$  are not known beforehand, it is natural to treat them as a random discrete distribution with a Dirichlet process prior. The  $a_i^i$  reflect the dissimilarity of fraction  $i$  from the other fractions by controlling the rate of sampling peptides directly from the root distribution  $G$ . We account for their distinguished role by putting prior weight  $\alpha_a^i$  on  $a_i^i$  and incorporating this parameter by assuming for the  $a^i$  a biased (in the sense of [6]) Dirichlet process prior  $\text{DP}_i(\gamma_a^i, \alpha_a^i, M)$  with uniform base measure  $M := (1/I)_{1..I}$ . In the following, we will refer to the  $a^i$  as the adapter distributions.

The self-referential base measures  $A_j^i$  are a crucial component of this process since they capture the important overlap of peptide distributions across the fractions  $j$  arising in a multidimensional fractionation experiment. The step from the simple base measure  $G$  as described in [6] to the self-referential base measure enables to appropriately characterize the peptide distributions describing such an experiment.

Putting together the precedent considerations we fully characterize the stochastic source of a in a multidimensional fractionation experiment by

$$\begin{aligned}
 G & \mid \gamma_r, d_r, H & \sim & \text{PY}(\gamma_r, d_r, H) \\
 a^i & \mid \gamma_a^i, \alpha_a^i, M & \sim & \text{DP}_i(\gamma_a^i, \alpha_a^i, M) \\
 G_j^i & \mid \gamma_c^i, A_j^i & \sim & \text{DP}(\gamma_c^i, A_j^i) \\
 \pi_t & \mid \pi_{t-1}^i = j & \sim & G_j^i
 \end{aligned} \tag{5}$$

Note that it is straightforward to assume Pitman-Yor process priors for all distributions. This choice though comes at the cost of additional parameters that have to be learned from data. In this work we wanted to focus on robustness and therefore we decided to keep the priors of the child distributions as simple as possible.

### 2.3 Sampling Sequences of Protein Identifications

This section describes a nested, recursive *Chinese Restaurant* construction to sample peptides from the hierarchical process model with self-referential base measures given an already observed series  $\pi$  of already observed peptides, i.e. how to simulate further experiments.

For each distribution in the hierarchical process model we have a restaurant representation, i.e. a seating arrangement. Specifically, we denote the restaurants corresponding to the  $G_j^i$  as  $R_{ij}^c = (n_{ijk}^c)_{k=1}^K$ , those to the  $a^j$  as  $R_i^a = (n_{ii'}^a)_{i'=1}^I$  and the root restaurant as  $R^r = (n_k)_{k=1}^K$ . To keep the notation uncluttered we incorporate the prior weights  $\alpha_a^i$  into the counts  $n_{ii}^a$  and respectively  $R_i^a$ .  $\mathbf{R}$  denotes the set of all restaurants. Note that a set of seating arrangements  $\mathbf{R}$  implies a series  $\pi$  of observed identifications. We further summarize the set of parameters by  $\boldsymbol{\theta} := (\gamma_r, d_r, \gamma_a^1, \dots, \gamma_a^I, \gamma_c^1, \dots, \gamma_c^I)$ .

For a given set of seating arrangements  $\mathbf{R}$  we now want to sample the identification  $\pi_t$  for fraction  $i$  and preceding identification  $\pi_{t-1} = j$ . Verbally, we first have to iterate the *Chinese Restaurant* construction for the corresponding child distribution. In case this iteration triggers a sampling event of its base measure, we have to determine which of its mixture components is to be sampled. Therefore we iterate the *Chinese Restaurant* construction of the corresponding adapter distribution. Subsequently, either the root restaurant or, recursively, some of the sibling child restaurants of another fraction is iterated. This procedure can summarized as shown below.

```

SAMPLEIDENTIFICATION( $i, j, \mathbf{R}, \boldsymbol{\theta}, H, M$ )
1   $\pi \leftarrow \text{SEAT}(R_{ij}^c, \gamma_c^i, 0, 0)$  // sample child
2  if  $\pi = 0$ 
3    then  $i' \leftarrow \text{SEAT}(R_i^a, \gamma_a^i, 0, M)$  // sample adapter
4      if  $i' \neq i$ 
5        then  $\pi \leftarrow \text{SAMPLEIDENTIFICATION}(i', j, \mathbf{R}, \boldsymbol{\theta}, H, M)$ 
6        else  $\pi \leftarrow \text{SEAT}(R^r, \gamma_r, d_r, H)$  // sample root
7  return  $\pi$ 

```



The nested, recursive *Chinese Restaurant* construction serves to simulate further experiments, i.e. to sample more peptides given an already observed series  $\pi$  of peptides and will be useful in the following section to derive a likelihood function for parameter estimation.

### 2.4 Empirical Bayes Parameter Estimate

Parameters of the hierarchical process model from section 2.2 can be estimated from a series  $\pi$  of identifications by empirical Bayes inference, i.e. by choosing the parameters to maximize a likelihood function  $\mathcal{L}_{\hat{\mathbf{R}}}$ .

$$\hat{\theta} := \arg \max_{\theta} \mathcal{L}_{\hat{\mathbf{R}}}(\theta) \tag{6}$$

In the following we will specify  $\mathcal{L}_{\hat{\mathbf{R}}}$ . Sampling a series  $\pi$  of identifications reduces to iterate various *Chinese Restaurant* constructions according to the probabilities in (2). We can define a likelihood function  $\mathcal{L}_{\mathbf{R}}(\theta)$  for a set of seating arrangements  $\mathbf{R}$ , or the corresponding series  $\pi$  of identifications.

$$\mathcal{L}_{\mathbf{R}}(\theta) = \mathcal{L}_{\text{cr}}(R^r, \gamma_r, d_r) \cdot \prod_{i=1}^I \mathcal{L}_{\text{cr}}(R_i^a, \gamma_a^i) \cdot \prod_{j=1}^J \mathcal{L}_{\text{cr}}(R_{ij}^c, \gamma_c^i) \tag{7}$$

where  $\mathcal{L}_{\text{cr}}(R, \gamma, d) / \mathcal{L}_{\text{cr}}(R, \gamma)$  corresponds to the likelihood of achieving a seating arrangement  $R$  in a single restaurant representation of a Pitman-Yor/Dirichlet process sample with parameters  $\gamma, d/\gamma$ . Note that prior weights  $\alpha_a^i$  of the adapter processes are appropriately incorporated into  $R_i^a$  and they are therefore not explicitly listed.

$$\mathcal{L}_{\text{cr}}(R, \gamma, d) = \frac{\prod_{k=1}^K (\gamma + kd) \cdot \prod_{n=1}^{n_k} (n - d)}{\prod_{n=1}^N (n + \gamma)} \tag{8}$$

with  $N = \sum_{k=1}^K n_k$  and  $K$  corresponding to the number of populated tables.

We do observe the series  $\pi$  of identifications. Though we only have incomplete knowledge about  $\mathbf{R}$ . We observe the seating arrangements  $R_{ij}^c$  of the child processes.

$$n_{ijk}^c = |\pi_t^i : (\pi_{t-1}^i = j) \wedge (\pi_t^i = k)| \tag{9}$$

where the  $\pi_t^i \in \pi^i$  denote identifications observed exclusively in fraction  $i$ . We do not directly observe  $R^r$  and the  $R_i^a$ . We present a sparse estimate for  $\mathbf{R}$  that is consistent with  $\pi$  and complies with a minimal number of seating events in the root restaurant representation  $R^r$  of the root distribution  $G$ . Consider the representation matrix  $M$  with entries  $m_{ik}$  equalling one if a peptide  $k$  has been observed in fraction  $i$  or zero otherwise. We want each peptide discovery  $k$  to be represented by some fraction  $f_k$ . We further want to choose the number of representing fractions to be as small as possible. This problem is more commonly known as the NP-hard set cover problem [14]. We compute the  $f_k$  with the greedy

heuristic, choosing at each step the fraction which covers the largest number of remaining different peptides. Every time the peptide  $k$  is discovered, i.e. sampled for the first time in a child process, we choose the corresponding adapter process to trigger a sampling event in  $f_k$ . Accordingly, we estimate the hidden seating arrangements of the adapter and root restaurant representations.

$$\begin{aligned} n_{ii'}^a &= |i, j, k : (f_k = i') \wedge (\exists t : (\pi_{t-1}^i = j) \wedge (\pi_t^i = k))| \\ n_k^r &= |i, j, k : (f_k = i) \wedge (\exists t : (\pi_{t-1}^i = j) \wedge (\pi_t^i = k))| \end{aligned} \quad (10)$$

We finally determine the parameters  $\hat{\theta}$  by optimizing  $\mathcal{L}_{\hat{\mathbf{R}}}$  with a quasi-Newton method [15]. In summary, we obtain an empirical Bayes parameter estimate from an observed series  $\pi$  of identifications.

## 2.5 Proteome Coverage Prediction with False Identifications

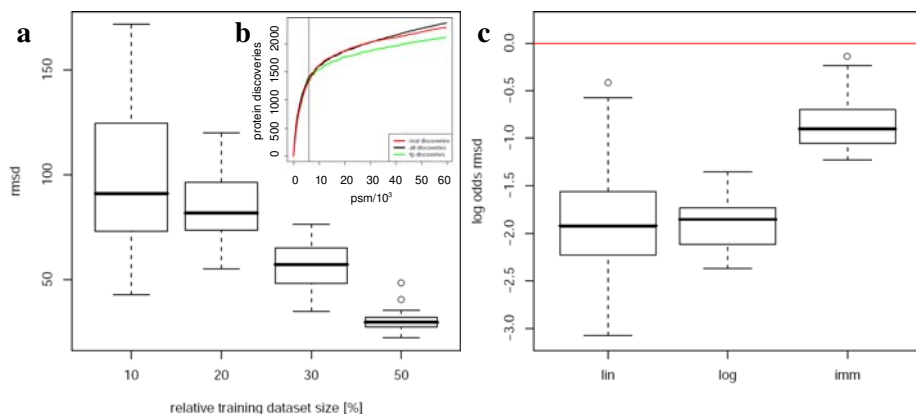
At this point we can specify how to predict the number of new peptide discoveries for future experiments from a series  $\pi$  of already observed identifications. In a first step, we estimate the parameters and hidden variables of the hierarchical process model (2.2) as described in the preceding section 2.4. Second, we sample  $m$  peptide series  $(\pi_{new,i})_{i=1}^m$  by means of the nested *Chinese Restaurant* construction (2.3). For each  $\pi_{new,i}$  we count the number of new discoveries. The expected proteome coverage we estimate as the mean of discovery counts across all  $\pi_{new,i}$ .

In practice, the series  $\pi$  of observed peptides corresponds to a series of peptide-spectrum matches that have been inferred computationally. Obviously peptide-spectrum matches are not perfect. Fortunately, the fraction of false positive peptide-spectrum matches is typically known [16,17]. Furthermore it has been observed that false positive peptide-spectrum matches distribute in a uniform-like manner across the protein database [6,10]. To account for false positive peptide-spectrum matches we adaptively estimate parameters and we adaptively sample novel peptide identifications as described in [6].

## 3 Results

We present results that demonstrate the proteome coverage prediction performance of our hierarchical process model. To this end we studied a large multi-dimensional fractionation experiment of a *L. interrogans* sample. We compared to a recent approach designed for (one dimensional) LC-MS/MS experiments [6] and to ad hoc extrapolation methods. We further extrapolated proteome coverage for the *L. interrogans* sample to make statements about maximal coverage.

We studied an integrated dataset acquired from multidimensional fractionation experiments for the bacterium *L. interrogans*. After protein extraction and tryptic digestion, the resulting peptide mixture was fractionated according to the isoelectric point of the peptides by off gel electrophoresis and each of



**Fig. 2.** Proteome coverage prediction performance by cross validation. Training datasets generated by subsampling the complete set of peptide-spectrum matches. Test of prediction performance on complete dataset. (a) Hierarchical process model accuracy in terms of root mean square deviation (rmsd) from the true progression of proteome coverage. Columns correspond to relative training dataset size compared to the complete *L. interrogans*. (b) Example trajectory for prediction from dataset instance with 10% relative size. Plot shows trajectory of observed (real), predicted true positive (tp) and including false positive protein discoveries (all). (c) Performance comparison of hierarchical process model with infinite Markov model (imm), extrapolation of logarithmic regression (log) and linear extrapolation of last experiment (lin). Box plot of log odds of rmsd ( $\log(\text{rmsd}_{\text{ref}}/\text{rmsd}_{\text{comp}})$ ) for reference and compared method (lin, log, imm). Median log odds for comparison with the other methods are significantly lower than zero, indicating weaker performance than our approach. The hierarchical process model is capable to reliably predict proteome coverage from a small amount of identifications and clearly outperforms other applicable methods.

the 24 fractions analyzed by LC-MS/MS coupled to a FT-LTQ high mass accuracy instrument. Target-decoy database search with Sequest/PeptideProphet [16] resulted in 59918 peptide-spectrum matches at a false discovery rate of 1% (Schmidt *et al.*, manuscript in preparation).

We assessed proteome coverage prediction performance in a cross validation scenario. Briefly, we generated various training datasets of decreasing size by subsampling the complete set of peptide-spectrum matches. We performed proteome coverage prediction for each training dataset and assessed accuracy by comparing to the real proteome coverage progression of the complete dataset. Precisely, we generated 20 training datasets by 20 times sampling 10% of all peptide-spectrum matches in the dataset while preserving their fraction association. We repeated this procedure by also sampling 20, 30 or 50% of all peptide-spectrum matches, finally obtaining 80 training datasets of varying size.

We assessed the prediction accuracy of the hierarchical process model (Fig. 2a). Prediction accuracy is measured as root mean square deviation of predicted and actually observed progression of proteome coverage. Proteome coverage

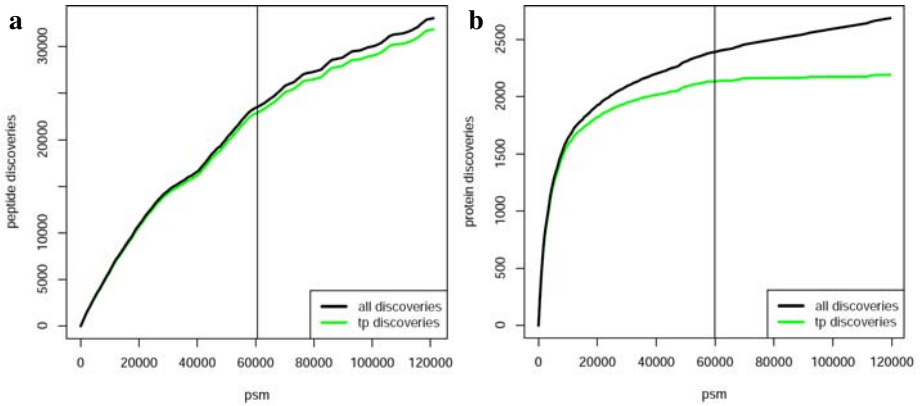
corresponds to number of protein discoveries. Prediction accuracy is reasonable already for the smallest training dataset sizes, i.e. 10% of the complete *L. interrogans* dataset. Fig. 2b depicts an example prediction for the set of smallest training datasets. As expected, prediction accuracy improves further for training datasets of larger size. Similar results are obtained for prediction of proteome coverage in terms of peptide discoveries (data not shown). We conclude that our approach is able to reliably predict proteome coverage already from a small amount of data.

We compared the hierarchical process model to other methods. We chose two simple general purpose extrapolation methods and a method designed for proteome coverage prediction of non-integrated datasets. We first considered an extrapolation scheme that linearly extrapolated proteome coverage progression of the last LC-MS/MS experiment of a training series. Second, we considered the extrapolation of a logarithmic regression ( $y = a \log x + b$ ). We assessed prediction performance on the 80 training series as described above and observed that the hierarchical process model clearly outperforms the other methods (Fig. 2c). These results indicate that proteome coverage prediction for integrated datasets is a non-trivial task that is not solved satisfactory by ad hoc extrapolation methods and is different from the related task of proteome coverage prediction for non-integrated datasets.

We estimated saturation proteome coverage for *L. interrogans* given the experimental workflow described above. Therefore we performed proteome coverage prediction for in silico repetition of all experiments. Proteome coverage in terms of peptide discoveries appears to steadily increase (Fig. 3a). Proteome coverage in terms of protein discoveries also seems to increase (Fig. 3b). This observation is however only true for all protein discoveries including the false positive ones. Since our approach separately accounts for the contribution of false and true positive protein discoveries (see section 2.5), we could exclusively monitor the progression of true protein discoveries. We observe that the number of true positive protein discoveries does not change significantly. Considering the rate of new true positive discoveries, we effectively have reached saturation coverage for *L. interrogans*.

## 4 Discussion

For the first time, we propose a method to predict proteome coverage for multidimensional fractionation experiments. This achievement is an important enabling step for experimentalists since multidimensional fractionation experiments so far have the largest potential to comprehensively characterize a proteome. We present a novel hierarchical process to characterize distributions arising in the course of these experiments. This approach conceptually extends methods exclusively suited for single fraction experiments [6], by introducing self-referential base measures that accommodate similarities among different experiment fractions. Our approach is generic since it operates on the level of peptide or protein distributions and, therefore, it conceptually accommodates any kind of heterogeneous set of fractions being analyzed by LC-MS/MS. Fractions do not necessarily



**Fig. 3.** Proteome coverage prediction beyond the *L. interrogans* dataset. Vertical lines denote the extent of the dataset in terms of acquired peptide-spectrum matches (psm). Trajectories correspond to predicted true positive (tp) and including false positive discoveries (all) (a) Progression of peptide discoveries. (b) Progression of protein discoveries. Protein discovery rate stagnates compared to the steadily increasing number of peptide discoveries. The *L. interrogans* dataset achieves saturation coverage at the level of protein discoveries.

have to originate from a single fractionation experiment. The considered fractions might also be derived from different tissues or cell cultures as long as their analysis is based on the same sequence database. Although we explicitly describe an approach that accounts for two fractionation steps, it is conceptually straightforward to extend it from a two level to a higher level hierarchy. However, the corresponding experimental setups are rarely encountered in practice. We show that our model reliably predicts proteome coverage of future experiments from a small amount of already performed experiments and clearly outperforms other methods.

Besides providing predictions at the level of peptide discoveries, we demonstrate that our approach yields reliable predictions of proteome coverage in terms of protein discoveries. Specifically, we require the set of considered fragment ion spectra to be unambiguously assigned to a protein identity to estimate future proteome coverage. This requirement is usually met, since possible ambiguities introduced by peptide-spectrum matches whose sequence maps to several protein identities are typically resolved by protein inference engines, e.g. by reporting a minimal consistent set of protein identifications [18]. It will though be interesting to extend our approach to allow for ambiguity in the protein identity assignments.

There has been considerable discussion in the past about when to consider a proteome to be mapped out. Our approach to proteome coverage prediction enables us to detect saturation coverage for any kind of shotgun proteomics dataset. In this study the *L. interrogans* dataset reaches saturation coverage at the level of protein discoveries. Out of 3740 proteins reported in the sequence

database, roughly 2000 proteins can be faithfully observed — not less but also not a lot more. This analysis is a remarkable result considering the manageable amount of experimentation (24 LC-MS/MS runs). It should be noted that this result is valid for the given experimental setup, such as type of protein extraction, enzymatic digestion, fractionation method, type of mass spectrometer. Despite the sensitive state-of-the-art approach reported here, it remains conceivable that other experimental approaches turn out to be able to explore other parts of the *L. interrogans* proteome. Their potential could though be evaluated with the hierarchical process model presented here. Therefore the presented method is suited to assist method development since it objectively assesses the potential of a particular method to explore a proteome.

Characterizing more complex proteomes (e.g. human) necessitates a considerably larger amount of experimentation. In this context it will be promising to perform proteome coverage prediction for different experimental strategies at an early stage of the project to design future experiments such that maximal proteome coverage is achieved efficiently. Our approach enables for the first time to accommodate any multidimensional fractionation strategy to perform this task. Efficient study design will help to save costly experiments, contribute to the reliability of the final set of protein discoveries [6,10] and furthermore enhance subsequent directed/targeted proteomics studies [19,20].

## Acknowledgments

We thank Alexander Schmidt and Lukas Reiter for carefully reading the manuscript. We further thank Alexander Schmidt for kindly providing the *L. interrogans* data set. The project was supported in part by internal funds from ETH Zurich and by SystemsX.ch, the Swiss initiative for systems biology.

## References

1. Domon, B., Aebersold, R.: Mass spectrometry and protein analysis. *Science* 312(5771), 212–217 (2006)
2. Nesvizhskii, A.I., Vitek, O., Aebersold, R.: Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nat. Methods* 4(10), 787–797 (2007)
3. Brunner, E., Ahrens, C.H., Mohanty, S., Baetschmann, H., Loevenich, S., Potthast, F., Deutsch, E.W., Panse, C., de Lichtenberg, U., Rinner, O., Lee, H., Pedrioli, P.G., Malmstrom, J., Koehler, K., Schrimpf, S., Krijgsveld, J., Kregenow, F., Heck, A.J., Hafen, E., Schlapbach, R., Aebersold, R.: A high-quality catalog of the *Drosophila melanogaster* proteome. *Nat. Biotechnol.* 25(5), 576–583 (2007)
4. Eriksson, J., Fenyo, D.: Improving the success rate of proteome analysis by modeling protein-abundance distributions and experimental designs. *Nat. Biotechnol.* 25(6), 651–655 (2007)
5. Beal, M., Ghahramani, Z., Rasmussen, C.: The infinite hidden Markov model. *Advances in Neural Information Processing Systems* 1, 577–584 (2002)
6. Claassen, M., Aebersold, R., Buhmann, J.M.: Proteome coverage prediction with infinite Markov models. *Bioinformatics* 25(12), i154–i160 (2009)

7. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101(476), 1566–1581 (2006)
8. Teh, Y.W.: A hierarchical Bayesian language model based on Pitman-Yor processes. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pp. 985–992 (2006)
9. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability* 25(2), 855–900 (1997)
10. Reiter, L., Claassen, M., Schrimpf, S.P., Jovanovic, M., Schmidt, A., Buhmann, J.M., Hengartner, M.O., Aebersold, R.: Protein Identification False Discovery Rates for Very Large Proteomics Data Sets Generated by Tandem Mass Spectrometry. *Mol. Cell Proteomics* 8(11), 2405–2417 (2009)
11. Blackwell, D., MacQueen, J.B.: Ferguson distributions via poly urn schemes. *The Annals of Statistics* 1(2), 353–355 (1973)
12. Pitman, J.: Combinatorial stochastic processes. Technical Report 621, Dept. Statistics, U.C. Berkeley (2002)
13. Antoniak, C.E.: Mixtures of Dirichlet processes with applications to Bayesian non-parametric problems. *The Annals of Statistics* 2(6), 1152–1174 (1974)
14. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York (1972)
15. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2005) ISBN 3-900051-07-0
16. Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R.: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* 74(20), 5383–5392 (2002)
17. Elias, J.E., Gygi, S.P.: Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods* 4(3), 207–214 (2007)
18. Nesvizhskii, A.I., Keller, A., Kolker, E., Aebersold, R.: A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* 75(17), 4646–4658 (2003)
19. Schmidt, A., Gehlenborg, N., Bodenmiller, B., Mueller, L.N., Campbell, D., Mueller, M., Aebersold, R., Domon, B.: An Integrated, Directed Mass Spectrometric Approach for In-depth Characterization of Complex Peptide Mixtures. *Mol. Cell Proteomics* 7(11), 2138–2150 (2008)
20. Lange, V., Malmstrom, J.A., Didion, J., King, N.L., Johansson, B.P., Schafer, J., Rameseder, J., Wong, C.H.o., Deutsch, E.W., Brusniak, M.Y., Buhlmann, P., Bjorck, L., Domon, B., Aebersold, R.: Targeted Quantitative Analysis of *Streptococcus pyogenes* Virulence Factors by Multiple Reaction Monitoring. *Mol. Cell Proteomics* 7(8), 1489–1500 (2008)

# Discovering Regulatory Overlapping RNA Transcripts

Timothy Danford<sup>1</sup>, Robin Dowell<sup>1,\*</sup>, Sudeep Agarwala<sup>2</sup>,  
Paula Grisafi<sup>2</sup>, Gerald Fink<sup>2</sup>, and David Gifford<sup>1</sup>

<sup>1</sup> Massachusetts Institute of Technology

<sup>2</sup> Whitehead Institute

**Abstract.** STEREO is a novel algorithm that discovers cis-regulatory RNA interactions by assembling complete and potentially overlapping same-strand RNA transcripts from tiling expression data. STEREO first identifies coherent segments of transcription and then discovers individual transcripts that are consistent with the observed segments given intensity and shape constraints. We used STEREO to identify 1446 regions of overlapping transcription in two strains of yeast, including transcripts that comprise a new form of molecular toggle switch that controls gene variegation.

## 1 Introduction

Evidence has recently emerged from high-throughput expression datasets that overlapping RNA transcripts can play an important role in gene regulation. For example, an antisense transcript can be used to regulate its corresponding sense gene [8,3]. In budding yeast, a sense/antisense toggle has been shown to regulate the mating type of the cell [6]. The interference of a transcript on the same strand as a coding transcript is also sufficient to play a repressive role in the regulation of downstream genes [10,11].

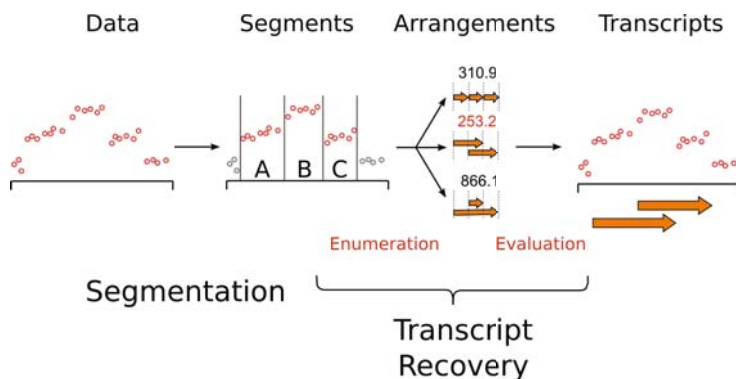
Discovering RNA transcript based cis-regulation requires the precise spatial localization of transcripts and the identification of their overlap with other, nearby transcripts. Contemporary algorithms for analyzing tiling microarray identify non-overlapping segments of coherent transcription [15], but they do not attempt to identify the transcripts that generated and potentially span the observed segments. A genomic locus which is multiply transcribed may produce a region of complex segmentation, but no additional resolution of such regions into separate, overlapping transcripts can be provided by dynamic programming or the probabilistic models used by segmentation algorithms.

We present a new algorithm, STEREO, for the computational analysis of overlapping transcription. STEREO is organized into two phases. The first phase implements segmentation and discovers genomic intervals which are transcribed in one of the input experiments. The identified genomic intervals are classified into transcript or background classes using both observed probe intensities and the

---

\* Currently: University of Colorado.





**Fig. 1.** Workflow description for segmentation and transcript recovery phases of the STEREO algorithm. The phases operate on sequence in a genomic region tiled by a microarray. The first phase partitions the genome into segments and assigns each segment a local transcription label. The second phase identifies clusters of transcription. For each cluster, transcript arrangements are enumerated and evaluated, and the optimal arrangement is chosen as the explanation for that cluster.

3' to 5' transcript intensity fall-off caused by reverse transcriptase processivity. The second phase, transcript reconstruction, resolves this labeled segmentation into consistent arrangements of explanatory RNA transcripts. STEREO performs a combinatorial search of all possible transcripts, given the constraints of transcript additivity and differential expression, to yield a segmentation.

STEREO is the first algorithm to computationally discover regions of complex transcription from segmentation, and to resolve those regions into overlapping transcripts. Overlapping transcripts fall into two mutually-exclusive categories, opposite and same-strand overlap, both of which may be expected to exhibit mutual interference or other regulatory properties. Opposite-strand overlap involves two transcripts transcribed from complementary DNA strands whose spatial extents overlap despite their different directions. Sense/antisense pairs of transcripts over the same gene are an example of opposite-strand overlap. Same-strand overlap occurs when two or more transcripts transcribed from overlapping portions of a DNA strand.

We tested STEREO on tiling expression data from two strains of yeast and discovered 1,446 instances of transcriptional overlap. Of these, 564 (39.0%) were overlapping in the same strand, a percentage consistent with previous estimates of alternate promoter usage in known yeast coding regions [12]. Northern blot analysis confirmed a same-strand interaction predicted by STEREO, and STEREO also identified opposite-strand transcripts that are organized into a novel form of molecular toggle switch [2] that controls the state of gene variegation.

The remainder of our paper is organized into sections that describe notation, previous work, and experimental design (Section 2), expression segmentation and results (Section 3), transcript discovery and results (Section 4), and a discussion (Section 5).

## 2 Preliminaries

### 2.1 Notation

We begin by outlining some basic notation for arrays and transcripts, shown in Table 1. For geometric descriptions of locations along the genome, we will use two terms: points and intervals. A point is the location of a single nucleotide in a genome assembly. Probes map to a genome assembly as point locations, based on the center of the interval to which their sequence is uniquely mapped. Intervals are convex subsets of the genome, single coherent loci specified completely by start and end positions.

**Table 1.** Array, segmentation, and transcript notation summary

$i, j, s, t$	probes $i$ , experiments $j$ , segments $s$ , transcripts $t$
$x_i$	genomic location of probe $i$
$y_{ij}$	intensity of probe $i$ in experiment $j$
$5'_{[s,t]}, 3'_{[s,t]}$	5' and 3' ends of segment $s$ or transcript $t$
$ x - x' $	linear distance along the genome, in bp
$t_s$	a label indicating the type of segment $s$
$\theta_s$	parameters of segment $s$
$\delta_{it}$	the distance $ x_i - 3'_t $ from probe $i$ to the 3' end of transcript $t$
$T_i$	set of transcripts that overlap probe $i$
$\gamma_{tj}$	intensity of transcript $t$ in experiment $j$
$\lambda_t$	3' log-linear slope of transcript $t$

A breakpoint set  $B = (b_1, \dots, b_N)$  is an ordered list of genomic locations which partition the genome into a set of non-overlapping intervals called segments. A segmentation is a set of segments which partition a complete genome. For a given set of breakpoints  $B$ , we use  $\mathbb{S}_B$  to indicate the segmentation defined by those breakpoints. If  $s \in \mathbb{S}_B$ , then  $s$  is a genomic interval whose endpoints ( $5'_s$  and  $3'_s$ ) are consecutive elements of the list  $B$ . A segmentation algorithm assigns each segment a type  $t_s$  and a set of parameters  $\theta_s$  which provide a local description of the probe values within that segment.

A transcript is a genomic interval, characterized by its start and end points  $5'_t$  and  $3'_t$ . It is a single, coherent message transcribed from the genome in one or more cells. It may be edited or it may be present in an unedited form, in which case it will appear as an interval when matched to the genome which produced it. Overlapping transcripts will produce complex regions of transcription. Transcribed regions are sections of the genome which may form part or all of a single mapped transcript or multiple adjacent and overlapping transcripts.

### 2.2 Prior Work

Analysis of tiling microarray data by segmentation was originally used for the analysis of comparative genomic hybridization [15,19]. Picard et al. described

the first dynamic-programming based segmentation algorithms for discovering regions of copy number variation in array-CGH experiments [15]. They later extended their algorithm to provide automatic labeling of segments using a hybrid dynamic programming/expectation maximization approach [16]. These methods derive their computational efficiency from the fundamental assumption that the segments they identify form a non-overlapping partition of the genome into spatially coherent intervals, an assumption which allows the use of dynamic programming approaches to discover optimal segmentations.

Tiling microarrays are also used to measure the transcription of genomic regions, and segmentation algorithms were similarly adapted to uncover consistently transcribed regions in those datasets [18]. Huber et al. adapted the segmentation algorithm of Picard to identify transcribed regions from tiling arrays [7]. This method was then used in David et al., which published the first genome-wide tiling microarray study of transcription in yeast [4]. One additional feature of tiling microarrays was their ability to discover strand-specific transcription through the use of strand-specific probes and experimental protocols which preserved the strand-specificity of the sample. The array results of David et al. were strand-specific, and so were able to identify regions of opposite-strand overlapping transcription.

Microarrays are not the only method for analyzing transcription on a genome-wide scale and in an unbiased manner; sequencing of cDNA has been a standard way to identifying unknown transcripts. Miura et al. sequenced expressed cDNA tags to produce a catalog of 5' and 3' transcript end-points throughout the yeast genome [12]. Sequencing measures single transcripts (and not transcribed regions) directly, and therefore can give information about the structure of transcript overlaps, starting, and ending points only if the read length is long enough relative to the transcript lengths.

The use of new, high-throughput short read sequencing machines to investigate transcription has led to the recent adoption of RNA-seq as a measurement of genome wide transcription [9,21]. RNA-seq experiments sequence fragments of transcripts which are randomly selected from the sample. Nagalakshmi et al. provided the first strand-insensitive view of transcription through RNA-seq in budding yeast [13]. These results have been extended in a strand-specific manner in related strains of yeast by Wilhelm et al. [20]. Unlike traditional sequencing, which produces longer reads, these unpaired-end short read sequencing techniques are unable to give us a full picture of the transcripts from which they were taken and suffer from the same problem of transcript mixture as microarrays. Some sequencing protocols produce reads which are insensitive to the strand of the underlying transcript, requiring that downstream computational analyses include strand-differentiation as one of their goals [14].

### 2.3 RNA *cis*-Regulation in S288C and $\Sigma$ 1278b

Using an array designed to probe the S288C genome at approximately 50 base-pair resolution, we designed a set of experiments intended to reveal differences in transcription regulation between two closely related strains of *Saccharomyces*

*cerevisiae*: S288C and  $\Sigma$ 1278b [5]. Each array had two channels, Cy3 and Cy5, which were used to simultaneously measure the expression in the two strains. In addition to the haploid (mat- $\alpha$ ) dataset of [5], we generated diploid expression in rich media with a technical replicate of each experiment. Treating each channel of each array as a separate logical experiment, this design provided us with eight total experiments on which to perform our segmentation and analysis. Data was normalized across experiments using quantile normalization [1].

### 3 Segmenting Expression Using Multiple Constraints

The segmentation phase of our algorithm partitions the genome into a complete set of non-overlapping regions. Each block, or segment, in the partition is labeled either  $\mathbf{t}_s = \text{TRANSCRIBED}$  or  $\mathbf{t}_s = \text{BACKGROUND}$  and assigned a set of local parameters that model the microarray probe observations within the segment. The segmentation considers multiple microarray experiments as input and learns a single segmentation that jointly explains all the input experiments. Segments may be assigned local parameters on an experiment specific basis, but the locations of the segments and the breakpoints that divide them are common across all experiments. Each label (**TRANSCRIBED** and **BACKGROUND**) corresponds to a model class, each with different complexities (requiring a penalty for the choice of a more complex class). The algorithm chooses from two classes, a flat model class that fits a mean and a variance to a given segment and represents the **BACKGROUND** segment label, and a linear model class that fits a line to the log intensities of the probes in a segment and is used to model the **TRANSCRIBED** label. The linear model class captures the 3' falloff effect created by the reverse transcriptase step of our experimental protocol. Both model classes can be represented by their log likelihood functions:

$$\mathcal{L}_j^{(1)}(x_1, x_2, \mu_j, \sigma) = \frac{1}{2} \log(\sigma) \sum_{i: x_1 \leq x_i \leq x_2} \frac{(y_{ij} - \mu_j)^2}{2\sigma^2} \quad (1)$$

$$\mathcal{L}_j^{(2)}(x_1, x_2, \mu_j, \lambda, \sigma) = \Pi + \frac{1}{2} \log(\sigma) \sum_{i: x_1 \leq x_i \leq x_2} \frac{(y_{ij} - \log(\mu_j e^{\delta_i \lambda}))^2}{2\sigma^2} \quad (2)$$

Here the  $x_1$  and  $x_2$  parameters are the bounds of the segment, while  $x_i$  and  $y_{ij}$  are the location of probe  $i$  and the value of probe  $i$  in experiment  $j$ , respectively.  $\Pi$  is a penalty term which corrects for the choice of the more complex (linear) model class, and is set through training against synthetically generated data. For a fixed pair of segment bounds, the choice of parameters for either model class are obtained by maximizing the corresponding log likelihood functions  $\mathcal{L}^{(1)}$  or  $\mathcal{L}^{(2)}$ . For either likelihood function, we will write  $\theta^* \equiv \arg \max_{\theta} \sum_j \mathcal{L}_j(x_1, x_2, \theta)$  to indicate the maximum likelihood values of the parameters given the segment boundaries  $x_1$  and  $x_2$ , and  $\mathbb{L}(x_1, x_2) \equiv \mathcal{L}(x_1, x_2, \theta^*)$  for the log-likelihood as a function of just the segment endpoints.

### 3.1 Segmentation Phase Uses Dynamic Programming to Find an Optimal Segmentation

The algorithm finds an optimal set of segmentation boundaries such that the total log likelihood of all probe observations from all experiments is maximized. Since a segmentation is a partition that separates the genome into non-overlapping regions, this can be accomplished through dynamic programming on the recursive formulation for  $\mathbb{L}$ .

$$\mathbb{L}(x_1, x_2) = \begin{cases} \mathbb{L}^{(1)}(x_1, x_2) \\ \mathbb{L}^{(2)}(x_1, x_2) \\ \max_{b \in [x_1, x_2]} \mathbb{L}(x_1, b) + \mathbb{L}(b, x_2) \end{cases} \quad (3)$$

The identity of any segment can be tracked by remembering which choice is maximizing. Those segments  $[x_1, x_2]$  for which the  $\mathbb{L}^{(1)}(x_1, x_2)$  is optimal are given the BACKGROUND label, while the TRANSCRIBED label is assigned to those for which  $\mathbb{L}^{(2)}(x_1, x_2)$  was optimal.

We also ran an implementation of the Picard segmentation algorithm on the S288C and  $\Sigma$ 1278b dataset. Although this method can be easily adapted to handle multiple experiments simultaneously, it lacks the ability to identify regions with a shape other than a flat regions of transcription; instead, it separates the sloped regions of transcription into “steps” of multiple flat segments. Therefore, the Picard algorithm is unable to handle a key feature of our experimental protocol (the 3' falloff) and unnecessarily single units of transcription into artificially complex sets of segments.

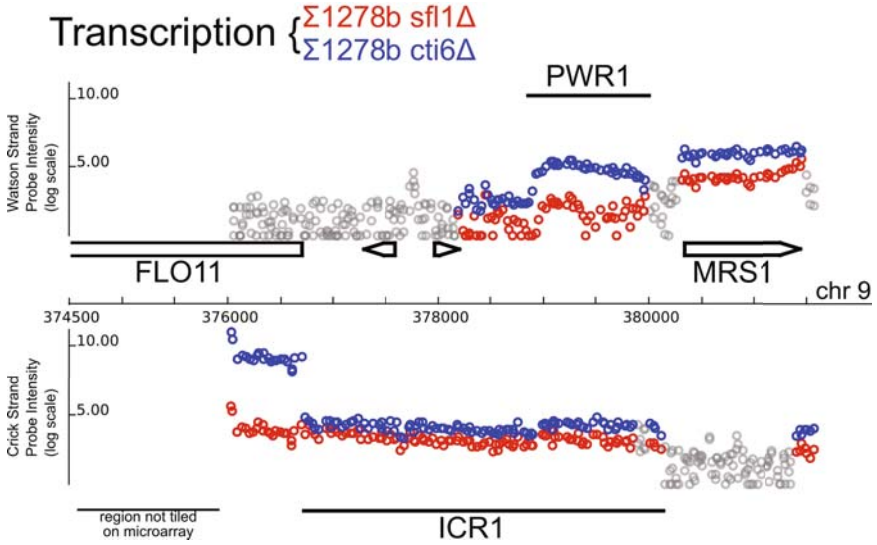
### 3.2 Segmentation Phase Discovers Novel Regulatory Transcription

We also examined the tiled expression data of Bumgarner [2], where  $\Sigma$ 1278b mutants were compared. Our segmentation algorithm, when run separately on each strand of this dataset, identified 14,076 segments on the Watson strand and 13,792 on the Crick strand. From the segmentation on the Watson strand, we identified 37.0% of the tiled genome as transcribed and from the Crick strand we identified 40.3%. Taken together, accounting for overlap, we identified 65.2% of the complete genome sequence as transcribed. The segmentation also recovered two noncoding transcripts whose regulatory function is related to their spatial overlap and interference with the production of a downstream coding transcript. In Figure 2, we show the locations of two noncoding transcripts, PWR1 and ICR1, that implement a new type of RNA molecular toggle [2].

## 4 STEREO Assembles Transcripts from Expressed Segments

### 4.1 An Additive Model for Overlapping Transcripts

Our model for overlapping transcripts employs two key constraints. First, we constrain same-strand overlapping transcripts that are co-expressed to display additive expression in their region of overlap. Second, we constrain transcripts



**Fig. 2.** We are able to discover noncoding transcription which is known to play a role in the regulation both of a downstream coding transcript (FLO11) and of each other. ICR1 and PRW1 are noncoding RNAs, reported in [2], whose regulatory function is related to their spatial overlap. The segmentation phase of our STEREO algorithm is able to find the complete PRW1 transcript and the 3' end of the ICR1 transcript in the Bumgarner dataset. Regions identified as BACKGROUND are shown in grey, TRANSCRIBED regions are shown in color. Genes are identified as arrowed boxes. The x-axis is genomic coordinates and the y-axis is log intensity.

to display 3' to 5' fall off in intensity corresponding to the processivity of reverse transcriptase in our experimental method. Our additivity constraint is reflected in the summation in Equation 4, and the slope constraint is reflected in the parameter  $\lambda$  that uniformly applies to all modeled transcripts. Equation 4 models observed intensities  $y_{ij}$  as the sum of transcript levels  $\gamma_{tj}$  associated with a particular transcript  $t$  in experiment  $j$ . Equation 4 makes the assumption that the noise of the array is log-normal, but that the transcripts themselves are additive in the non-logarithmic-scale of the array.

$$y_{ij} = \log\left(\sum_{t \in T_i} \gamma_{tj} e^{\lambda \delta_{it}}\right) + e_{ij} \tag{4}$$

If we give the unit level error term a probability distribution,  $e_{ij} \sim \mathcal{N}(\cdot; 0, \sigma_y)$ , we turn Equation 4 into a probabilistic model with log-likelihood function:

$$\mathcal{L}(\Gamma, \sigma, \lambda) = -N\sigma - \sum_i \sum_j \frac{(y_{ij} - \log(\sum_{t \in T_i} \gamma_{tj} e^{\lambda \delta_{it}}))^2}{2\sigma^2} \tag{5}$$

The vector of transcript intensities  $\Gamma = \{\gamma_{tj}\}$ , along with the transcript slope  $\lambda$  and probe level variance  $\sigma$ , are chosen to maximize the log likelihood function

in Equation 5. Since this equation is a non-linear function of a sum, there is not a simple closed-form solution for the maximizing parameters. Instead, we compute the derivatives of the log-likelihood function and maximize numerically using gradient ascent.

## 4.2 Enumerating and Evaluating Overlapping Transcripts

A maximum likelihood solution to Equation 5 provides a method for finding local parameters for a set  $T$  of overlapping transcripts. However, it does not answer the question of how we determine  $T$ . A poor choice of  $T$  will lead to estimates of transcript intensities that do not correspond to biological reality.

STEREO uses an enumeration-based search method to choose the transcript arrangement  $T$  which best explains the transcribed regions that are provided as input by the segmentation and labeling phase. We assume that the segmentation provided to the transcript discovery phase has correctly identified the starts and ends of transcripts as breakpoints in the segmentation, and correctly labeled each segment as transcribed or noise. Furthermore, we assume that every transcribed segment must be explained by at least one transcript, while noise segments will not be explained by any transcript.

We break the problem of transcript calling into independent sub-problems, called clusters. Each cluster is a spatially-consecutive sequence of transcribed segments, separated from every other cluster by one or more noise segments, or a chromosome boundary.

The STEREO algorithm first identifies the clusters corresponding to the input segmentation of tiling microarray data. Then for each cluster, it enumerates all possible transcript arrangements. Each cluster will have a finite number of arrangements, since there are a finite number of breakpoints in the cluster and we assume that the total number of transcripts does not exceed the number of segments in the cluster. For each enumerated transcript arrangement  $T$ , we find an optimal set of parameters  $\Theta_T \equiv \langle \Gamma_T, \lambda_T, \sigma_T \rangle = \arg \max \mathcal{L}(\Gamma, \lambda, \sigma)$  by maximizing the log-likelihood equation of the probes within the cluster. The penalized log-likelihood  $\mathcal{L}(\Gamma_T, \lambda_T, \sigma_T) - C(T)$  then provides a score by which to evaluate the fit of the transcript arrangement  $T$  to the cluster. The complexity penalty  $C(T) = \alpha|T| + \beta c_{\text{cover}}(T)$  assesses a constant penalty for the total number of transcripts in  $T$  and for the number of overlaps  $c_{\text{cover}}(T)$  in the arrangement. The penalties ( $\alpha$  and  $\beta$ ) are chosen to optimize transcript discovery against synthetically-generated data.

## 4.3 STEREO Transcript Discovery Recovers Appropriate SER3/SRG1 Transcripts

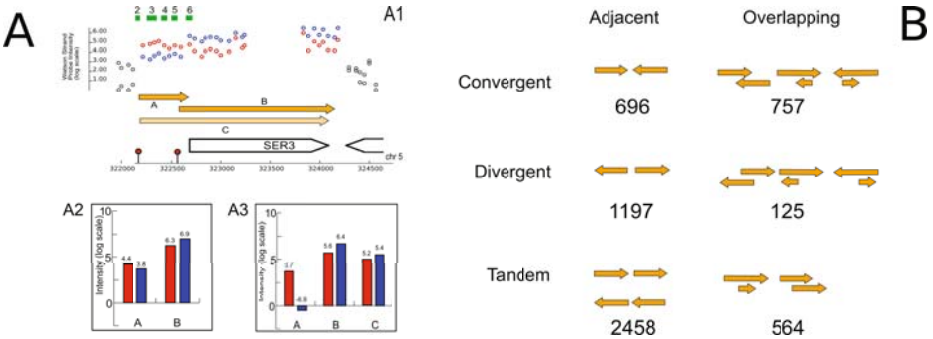
An example of known overlapping transcripts with regulatory interactions in yeast is the SER3 gene and its upstream intergenic transcript, SRG1. The SER3 gene is involved in serine biosynthesis and under repressing conditions its promoter is bound by significant levels of both TATA binding protein (TBP) and RNA polymerase II (Pol II). The expression of a short transcript that runs



through the SER3-proximal TATA element is associated with decreased expression of the SER3 transcript itself [10]. Furthermore, a nearly 2 kb read-through transcript starting from the SRG1 TATA element and extending through the entire SER3 gene itself was observed by northern analysis in the same study.

The SER3 and SRG1 genes, and their observed architecture of overlapping transcription, provide a convenient test of our ability to estimate relative intensities of overlapping transcripts. In Figure 3, we show that our tiling array data in S288C (red) and  $\Sigma$ 1278b (blue) around the SER3 and SRG1 locus. The figure depicts the locations of three overlapping transcripts, shown as orange arrows: one from the upstream SRG1 TATA element extending to the annotated start of the SER3 gene, the second from the SER3 TATA element extending to the end of the SER3 gene, and one 2 kb-long transcript starting from the SRG1 TATA element and extending through the SER3 gene itself.

Using our transcript intensity estimation method we reconstructed relative log-intensities of 4.4 and 6.3 for the A and B transcripts respectively; these values are consistent with previously reported concentrations for SRG1 and SER3 respectively [10]. Moreover, the fitted intensities are anti-correlated across cell types, between the two measured strains of yeast. When the SRG1 transcript drops the SER3 transcript rises, consistent with the claim that SRG1’s transcription represses that of SER3.



**Fig. 3.** **A** Reconstruction of transcript intensities at the SER3/SRG1 locus. **A1.** Probes included in either the SER3 or SRG1 region and in this analysis are displayed in either red (S288C) or blue ( $\Sigma$ 1278b) dots. Original probes from Martens et al. enriched for the SRG1 transcript are green boxes. Putative transcripts A, B, and C are shown in orange arrows and TATA elements with red dots. Transcript A corresponds to Martens SRG1 transcript while transcript B corresponds to SER3 transcript. Transcript C is the “readthrough” transcript Martens detected, extending exactly 2 kb. Transcript intensity analyses were carried out for two arrangements, (**A2**) just the A and B transcripts and (**A3**) all three transcripts. Each transcript has reconstructed intensities for both S288C (red) and  $\Sigma$ 1278b (blue) experimental data. **B** Schematics for classification of transcript pairs, along with the total number of cases STEREO identified, within each category.



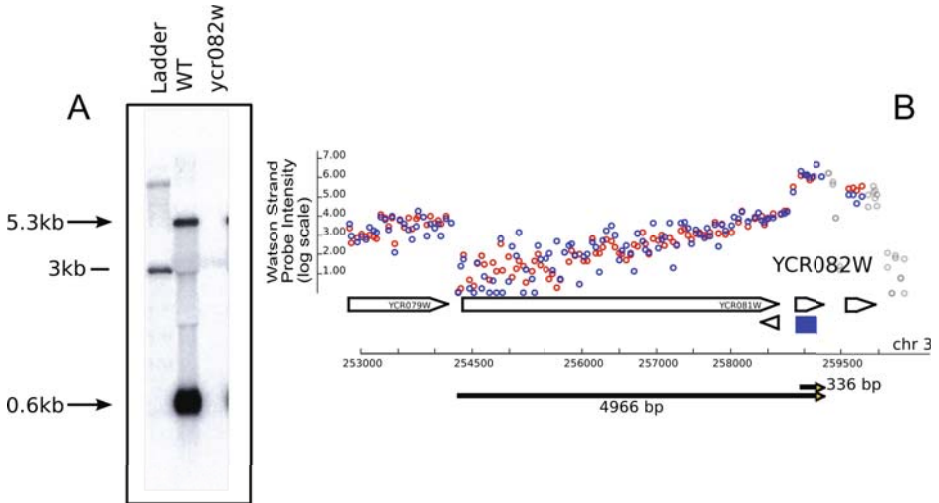
#### 4.4 Identification of 1446 Overlapping Transcripts

STEREO resolved the collected S288C and  $\Sigma$ 1278b expression datasets into 6609 transcripts. Most transcripts (5233) inferred by our method were strand singletons, covering a single region without a second overlapping transcript. However, our algorithm identified 1446 regions of overlapping transcription, of which 564 were same-strand overlapping transcripts. Figure 3 Part B shows a classification of transcript pairs into six categories depending on their relative orientation and overlap, and gives the number of transcript pairs that fell into each category from our dataset.

The segmentation and labeling phase has also been able to uncover overlapping transcript pair predictions which show differential expression between different cell types and strains, and whose variation is consistent with potential repressive regulatory interactions between the overlapping transcripts.

#### 4.5 Northern Analysis of Overlapping Predictions

In order to confirm one of our predictions we chose three of the predictions made by our algorithm to test with northern blot analysis. To facilitate northern blot analysis we chose examples to test that had a larger outer transcript with a smaller inner contained transcript that would readily be apparent in the



**Fig. 4.** Northern analysis was performed at YCR082W to test for the presence of multiple overlapping transcripts. Probes were chosen to cover the first 400 bp of the gene, shown as a blue square. The blot (A) shows two transcripts with lengths approximately 5 kb and 600 bp. These transcripts correspond (B) to two overlapping transcripts called by the STEREO algorithm with lengths of approximately 5 kb and 300 bp. For clarity, only the Watson strand is shown. Probes in TRANSCRIBED regions are shown in color for S288C (red) and  $\Sigma$ 1278b (blue) data.

experimental result. In one of the three locations tested northern blot analysis showed same-strand overlapping transcription with transcript lengths matching those produced by STEREO. This validated locus, YCR082W, provides a new example of tandem overlapping transcripts previously unknown in the literature<sup>4</sup>. Instead of reporting overlapping transcripts in this location, an alternate explanation would have been two tandem transcripts aligned head-to-tail; in this case, the transcript discovery algorithm reconstructs the more complex overlap based on our prior distribution over transcript intensities and our belief that higher-intensity transcripts are less likely than lower ones.

Zheng et al. have previously attempted to quantify the intensities of multiple overlapping transcripts using a hierarchical Bayesian model [22]. Their approach is limited, however, to the quantification of transcript intensities whose locations have already been specified from gene annotations or an external datasource. Rochette et al. have reported a set of overlapping transcripts at a genome-wide level in the parasite *Leishmania* [17]. These transcripts were identified by experimental means (5'-RACE) in a genome significantly smaller than yeast, however, and do not represent a comprehensive computational approach to transcript discovery.

## 5 Discussion

Our STEREO algorithm contains several unique features. In the segmentation phase, we simultaneously incorporated multiple experiments and utilized the slope of the transcription data to identify transcribed segments. In the transcript discovery phase we employed both additive intensity and differential expression to evaluate likely configurations of transcripts.

STEREO also has certain limitations. While a 3' to the 5' intensity fall off provides a useful constraint, it also makes it more difficult to accurately locate the 5' ends of long, low-abundance transcripts. In addition, STEREO is sometimes unable to separate same-strand overlapping transcripts without differential expression between conditions or strains. In these cases, overlapping transcript calling depends on our prior distributions on transcript intensities. A better understanding of the distribution of transcript abundances will improve the accuracy of our transcript reassembly algorithm. The combinatorial architecture of gene regulation is in part implemented by RNA based cis-regulation. We are making our set of 1446 candidate interactions available for other investigators.

## References

1. Bolstad, B.: Probe Level Quantile Normalization of High Density Oligonucleotide Array Data. Technical report, Division of Biostatistics, University of California, Berkeley (2001)
2. Bumgarner, S.L., Dowell, R.D., Grisafi, P., Gifford, D.K., Fink, G.R.: Toggle involving *cis*-interfering noncoding RNAs controls variegated gene expression in yeast. *Proceedings of the National Academy of Sciences* 106(43), 18321–18326 (2009)

3. Camblong, J., Iglesias, N., Fickentscher, C., Dieppois, G., Stutz, F.: Antisense RNA Stabilization Induces Transcriptional Gene Silencing via Histone Deacetylation in *S. cerevisiae*. *Cell* 131, 706–717 (2007)
4. David, L., Huber, W., Granovskaia, M., Toedling, J., Palm, C.J., Bofkin, L., Jones, T., Davis, R.W., Steinmetz, L.M.: A high-resolution map of transcription in the yeast genome. *PNAS* 103(14), 5320–5325 (2006)
5. Dowell, R.D., Ryan, O., Jansen, A., Cheung, D., Agarwala, S., Danford, T.W., Bernstein, D., Rolfe, P.A., Fink, G.R., Gifford, D.K., Boone, C.: Genotype to Phenotype: A Comparison of Two Interbreeding Yeast Strains Reveals Complex Genetics of Conditional Essential Genes (in submission)
6. Hongay, C., Grisafi, P., Galitski, T., Fink, G.: Antisense transcription controls cell fate in *Saccharomyces cerevisiae*. *Cell* 127(4), 735–745 (2006)
7. Huber, W., Toedling, J., Steinmetz, L.: Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics* 22(16), 1963–1970 (2006)
8. Hughes, T.A.: Regulation of gene expression by alternative untranslated regions. *Trends in Genetics* 22(3), 119–122 (2006)
9. Marioni, J., Mason, C., Mane, S., Stephens, M., Gilad, Y.: RNA-Seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research* 18, 1509–1517 (2008)
10. Martens, J.A., Laprade, L., Winston, F.: Intergenic transcription is required to repress the *Saccharomyces cerevisiae* SER3 gene. *Nature* 429, 571–574 (2004)
11. Martianov, I., Ramadass, A., Barros, A.S., Chow, N., Akoulitchev, A.: Repression of the human dihydrofolate reductase gene by a non-coding interfering transcript. *Nature* 445, 666–670 (2007)
12. Miura, F., Kawaguchi, N., Sese, J., Toyoda, A., Hattori, M., Morishita, S., Ito, T.: A large-scale full-length cDNA analysis to explore the budding yeast transcriptome. *PNAS* 103(47), 17486–17851 (2006)
13. Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., Snyder, M.: The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320, 1344–1349 (2008) 1158441
14. Parkhomchuk, D., Borodina, T., Amstislavskiy, V., Banaru, M., Hallen, L., Krobitsch, S., Lehrach, H., Soldatov, A.: Transcriptome analysis by strand-specific sequencing of complementary dna. *Nucleic Acids Research* (July 2009)
15. Picard, F., Robin, S., Lavielle, M., Vaisse, C., Daudin, J.-J.: A Statistical Approach for Array CGH Data Analysis. *BMC Bioinformatics* 6(27) (February 2005)
16. Picard, F., Robin, S., Lebarbier, E., Daudin, J.-J.: A segmentation/clustering model for the analysis of array CGH data. *Biometrics* 63, 758–766 (2007)
17. Rochette, A., Raymond, F., Ubeda, J.M., Smith, M., Messier, N., Boisvert, S., Rigault, P., Corbeil, J., Ouellette, M., Papadopoulou, B.: Genome-wide gene expression profiling analysis of leishmania major and leishmania infantum developmental stages reveals substantial differences between the two species. *BMC Genomics* 9(255) (2008)
18. Royce, T., Rozowsky, J., Bertone, P., Samanta, M., Stolc, V., Weissman, S., Snyder, M., Gerstein, M.: Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends in Genetics* 21(8), 466–475 (2005)
19. Urban, A.E., Korb, J.O., Selzer, R., Richmond, T., Hacker, A., Popescu, G., Cubells, J.F., Green, R., Emanuel, B.S., Gerstein, M.B., Weissman, S.M., Snyder, M.: High-resolution mapping of DNA copy alterations in human chromosome 22 using high-density tiling oligonucleotide arrays. *PNAS* 103(12), 4534–4539 (2006)

20. Wilhelm, B., Landry, J.R.: Rna-seq: quantitative measurement of expression through massively parallel rna-sequencing. *Methods* 48(3), 249–257 (2009)
21. Wilhelm, B., Marguerat, S., Watt, S., Schubert, F., Wood, V., Goodhead, I., Penkett, C., Rogers, J., Bahler, J.: Dynamic repertoire of a eukaryotic transcriptome surveyed at a single-nucleotide resolution. *Nature* 453, 1239–1243 (2008)
22. Zheng, S., Chen, L.: A hierarchical Bayesian model for comparing transcriptomes at the individual transcript isoform level. *Nucleic Acids Research*, 1–16 (2009)

# Alignment-Free Phylogenetic Reconstruction

Constantinos Daskalakis<sup>1</sup> and Sebastien Roch<sup>2</sup>

<sup>1</sup> CSAIL, MIT

<sup>2</sup> Department of Mathematics and Bioinformatics Program, UCLA

**Abstract.** We introduce the first polynomial-time phylogenetic reconstruction algorithm under a model of sequence evolution allowing insertions and deletions (or indels). Given appropriate assumptions, our algorithm requires sequence lengths growing polynomially in the number of leaf taxa. Our techniques are distance-based and largely bypass the problem of multiple alignment.

**Keywords:** Phylogenetic reconstruction, indels, alignment.

## 1 Introduction

We introduce a new efficient algorithm for the *phylogenetic tree reconstruction* (PTR) problem which rigorously accounts for insertions and deletions.

*Phylogenetic background.* A *phylogenetic tree* or *phylogeny* is a tree representing the speciation history of a group of organisms. The leaves of the tree are typically existing species. The root corresponds to their *most recent common ancestor* (MRCA). Each branching in the tree indicates a speciation event. It is common to assume that DNA evolves according to a Markovian substitution process on this phylogeny. Under such a model, a *gene* is a sequence in  $\{A, G, C, T\}^k$ . Along each edge of the tree, each site independently mutates according to a Markov rate matrix. The length of a branch is a measure of the amount of substitution along that branch<sup>1</sup>. The PTR problem consists in estimating a phylogeny from the genes observed at its leaves. We denote the leaves of a tree by  $[n] = \{1, \dots, n\}$  and their sequences by  $\sigma_1, \dots, \sigma_n$ .

The model of sequence evolution above is simplistic: it ignores many mutational events that DNA undergoes through evolution. At the gene level, the most important omissions are insertions and deletions of sites, also called *indels*. Stochastic models taking indels into account have long been known [1, 2], but they are not widely used in practice—or in theory—because of their complexity. Instead, most practical algorithms take a two-phase approach:

1. **Multiple sequence alignment.** Site  $t_i$  of sequence  $\sigma_i$  and site  $t_j$  of sequence  $\sigma_j$  are said to be *homologous* if they descend from the same site  $t_0$  of a common ancestor  $u$  *only through substitutions*. In the *multiple sequence alignment* (MSA) problem, we seek roughly to uncover the homology relation between  $\sigma_1, \dots, \sigma_n$ . Typically, the

---

<sup>1</sup> The precise definition of a branch length depends on the model of evolution. For roughly constant mutation rates, one can think of the branch length as proportional to the amount of time elapsed along a branch.

output is represented by a matrix  $\mathbf{D}$  of  $n$  aligned sequences of equal length with values in  $\{A, G, C, T, -\}$ . Each column of the matrix corresponds to homologous sites. The state  $-$  is called a *gap* and is used to account for insertions and deletions. For instance if sequence  $\sigma_i$  does not have a site corresponding to  $t_0$  in  $u$  above, then a gap is aligned with positions  $t_i$  of  $\sigma_i$  and  $t_j$  of  $\sigma_j$  (which belong to the same column).

2. **Phylogenetic tree reconstruction.** The matrix  $\mathbf{D}$  is then cleaned up by removing all columns containing gaps. Let  $\mathbf{D}'$  be this new matrix. A standard PTR algorithm is then applied to  $\mathbf{D}'$ . Note that substitutions alone suffice to explain  $\mathbf{D}'$ . (In fact, there are many other ways to deal with gaps but we do not describe them here.)

Traditionally, most of the research on phylogenetic methods has focused on the second phase.

In fact, current theoretical analyses of PTR assume that the MSA problem has been solved *perfectly*. This has been a long-standing assumption in evolutionary biology. But this simplification is increasingly being questioned in the phylogenetic literature, where it has been argued that alignment heuristics often create systematic biases that affect analysis [3,4]. Much recent empirical work has been devoted to the proper joint estimation of alignments and phylogenies [1,2,5,6,7,8,3,9]. Here, we give the first analysis of an efficient, provably consistent PTR algorithm in the presence of indels. Our new algorithm suggests that a rough alignment suffices for an accurate tree reconstruction—bypassing the computationally difficult multiple alignment problem.

*Theoretical properties of PTR.* In addition to computational efficiency, an important theoretical criterion in designing a PTR algorithm is the so-called *sequence-length requirement* (SLR). At a minimum, a reconstruction algorithm should be *consistent*, that is, assuming a model of sequence evolution, the output should be guaranteed to converge on the true tree as the sequence length  $k$  (the number of *samples*) goes to  $+\infty$  [10]. Beyond consistency, the sequence-length requirement (or convergence rate) of a PTR algorithm is the sequence length required for guaranteed high-probability reconstruction. The SLR is typically given as an asymptotic function of  $n$ , the number of leaves of the tree. Of course, it also depends on the substitution parameters.

A classical result due to Erdős et al. [11] states that, for general trees under the assumption that all branch lengths are bounded by constants, the so-called Short Quartet Method (SQM) has  $\text{poly}(n)$ -SLR. The SQM is a particular PTR algorithm based on estimating evolutionary distances between the leaf taxa, that is, the sum of the branch lengths between species. Such algorithms are known as *distance-based methods*. The basic theoretical result behind distance-based methods is the following: the collection of pairwise evolutionary distances between all species forms a special metric on the leaves known as an additive metric; under mild regularity assumptions, such a metric *characterizes* the underlying phylogeny interpreted as an edge-weighted tree, that is, there is a one-to-one correspondence between additive metrics and phylogenies; moreover, the mapping between them can be computed efficiently [12].

*A new approach.* In the classical theoretical setting above where the MSA problem is assumed perfectly solved (we refer to this setting below as the ESSW framework), the evolutionary distance between two species is measured using the Hamming distance (or

a state-dependent generalization) between their respective sequences. It can be shown that after a proper correction for multiple substitutions (which depends on the model used) the expectation of the quantity obtained does satisfy the additive metric property and can therefore serve as the basis for a distance-based PTR algorithm.

Moving beyond the ESSW framework, it is tempting to account for indels by simply using edit distance instead of the Hamming distance. Recall that the *edit distance* or *Levenshtein distance* between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. However, no analytical expression is known for the expectation of edit distance under standard indel models and computing such an expression appears difficult—if at all possible. An alternative idea is to compute the *maximum likelihood estimator* for the time elapsed between two species given their sequences. But this involves solving a nonconvex optimization problem and the likelihood is only known to be efficiently computable under a rather unrealistic assumption known as reversibility [11] (see below).

We use a different approach. We divide the sequences into quantile blocks (the first  $x\%$ , the second  $x\%$ , etc.). We show that by appropriately choosing  $x$  above we can make sure that the blocks in different sequences essentially “match” each other, that is, they are made of mostly homologous sites. We then compare the state frequencies in matching blocks and build an additive metric out of this statistic. As we show below, this is in fact a natural generalization of the Hamming estimator of the ESSW framework. However, unlike the Hamming distance which can easily be analyzed through standard concentration inequalities, proving rigorously that our approach works involves several new technical difficulties. Once a distance estimate is obtained, we use a standard distance-based reconstruction method. (No new algorithm is presented here.)

*Related work.* For more background on models of molecular evolution and phylogenetics, see, e.g., [13, 12, 14]. Following the seminal results of [11], there has been much work on sequence-length requirement, including [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30].

The multiple sequence alignment problem as a combinatorial optimization problem (finding the best alignment under a given pairwise scoring function) is known to be NP-hard [31, 32]. Most heuristics used in practice, such as CLUSTAL [33], MAFFT [34], and MUSCLE [35], use the idea of a guide tree, that is, they first construct a very rough phylogenetic tree from the data (using for instance edit distance as a measure of evolutionary distance), and then recursively construct alignments over subsets of species produced by “aligning alignments.” We point out that multiple sequence alignments are useful in their own right—not only to reconstruct phylogenies. In fact, it may be possible to use an approach similar to ours to construct good guide trees, as described above, and potentially obtain better alignments as a result.

To our knowledge, little theoretical work has been dedicated to the joint estimation of alignments and phylogenies, with the exception of Thattai [36] who gave consistency results for the reversible case in the limit where the deletion-to-insertion ratio tends to 1. However, no sequence-length requirement is obtained in [36]. In recent related work, Andoni et al. [37] considered the problem of reconstructing ancestral sequences in the presence of indels.

There is a large body of empirical research on alignment-free phylogenetic reconstruction, particularly in the context of genome-wide analyses. Such methods often work by counting  $k$ -mers (and variants). See e.g. [38] and references therein. The method we present here is closely related to  $k$ -mer statistics—in essence, we compute a 1-mer statistic, but we do so on a large number of roughly aligned blocks in order to derive more statistical power. In fact, our results can be seen as a first attempt to analyze rigorously this type of alignment-free approach.

## 1.1 Model of Sequence Evolution

*Phylogeny.* A *phylogeny* is represented by a binary tree  $T = (V, E)$ , whose leaves  $L \subset V$  correspond to extant species, and whose bifurcations denote evolutionary events whereby two new species are generated from an ancestor. The root of the phylogeny, denoted by  $r(T)$ , represents the common ancestor of all the species in the phylogeny, and we assume that all edges of  $T$  are directed away from  $r(T)$ ; so, if  $e = (u, v)$  is a branch of the phylogeny,  $u$  is the *parent* of  $v$  and  $v$  is the *child* of  $u$ . Moreover, if  $v'$  is in the subtree of  $T$  rooted at  $u$ , we call  $v'$  a *descendant* of  $u$  and  $u$  an *ancestor* of  $v'$ .

Along each branch of the phylogeny, the genetic material of the parent species is subject to modifications that produce the genetic material of its child species. A common biological assumption is that the genetic material of each species  $u$  can be represented by a binary sequence  $\sigma_u = (\sigma_u^1, \dots, \sigma_u^{K_u})$  of length  $K_u$  over a finite alphabet—we work here with the binary alphabet  $\{0, 1\}$  for simplicity—and that the changes to which  $\sigma_u$  is subjected along the branch  $e = (u, v)$  are described by a Markov process. In particular, the Markov property implies that, given the sequence  $\sigma_u$  of  $u$ , the sequence  $\sigma_v$  is independent of the sequences of the species outside the subtree of  $T$  rooted at  $u$ .

A simplifying assumption commonly used in phylogenetics is that all species have sequences of the same length and, moreover, that every *site*, i.e., every coordinate, in their sequences evolves independently from every other site. In particular, it is assumed that, along each branch  $e = (u, v)$  of the phylogeny, every site  $\sigma_u^j$  of the sequence  $\sigma_u$  is flipped with probability  $p_e$  to the value  $1 - \sigma_u^j$  independently from the other sites. This model is known as the Cavender-Farris-Neyman (CFN) model. A simple generalization to  $\{A, G, C, T\}$  is known as the Jukes-Cantor (JC) model. See, e.g., [14].

*Accounting for indels.* In this paper, we consider a more general evolutionary process that accounts for the possibility of insertions and deletions. Our model is similar to the original TKF91 model [11], except that we do not enforce reversibility. In our model, every edge  $e = (u, v)$  of the phylogeny is characterized by a quadruple of parameters  $(t_e; \eta_e, \mu_e, \lambda_e)$ , where  $t_e$  is the evolutionary time between the species  $u$  and  $v$ , and  $\eta_e$ ,  $\mu_e$  and  $\lambda_e$  are respectively the *substitution*, *deletion* and *insertion* rates. The process

<sup>2</sup> We can also consider richer alphabets, e.g.,  $\{A, C, G, T\}$ , without much modification. See journal version.

<sup>3</sup> We do not use an immortal link and we do not assume that the length process is at stationarity. Our techniques can also be applied to the TKF91 model without much modifications. We leave the details to the reader.



by which the sequence at  $v$  is obtained from the sequence at  $u$  is defined below. (The process can be simply described as a continuous-time Markov process [39]. We give a full description for clarity.)

**Definition 1 (Evolutionary Process on a Branch).** *Given an edge  $e = (u, v)$ , with parameters  $(t_e; \eta_e, \mu_e, \lambda_e)$ , the sequence  $\sigma_v$  at  $v$  is obtained from the sequence  $\sigma_u$  at  $u$  according to the following Markov procedure:*

1. initialize  $\sigma_v := \sigma_u, K_v := K_u$  and  $t_\ell := t_e$ ;  
*/\* $t_\ell$  is the remaining time on the edge  $e$ \*/*
2. while  $t_\ell > 0$ 
  - let  $I_0, I_1, \dots, I_{K_v}$  be exponential random variables with rate  $\lambda_e, D_1, \dots, D_{K_v}$  exponential random variables with rate  $\mu_e$ , and  $M_1, \dots, M_{K_v}$  exponential random variables with rate  $\eta_e$ ; suppose that these random variables are mutually independent and let  $\mathcal{T}$  be their minimum;
  - if  $\mathcal{T} > t_\ell$  break; otherwise: if  $I_j = \mathcal{T}$ , insert a new site whose value is chosen uniformly at random from  $\{0, 1\}$  between the sites  $\sigma_v^j$  and  $\sigma_v^{j+1}$  of  $\sigma_v$ ; <sup>4</sup> if  $D_j = \mathcal{T}$ , delete the site  $\sigma_v^j$  from  $\sigma_v$ ; and if  $M_j = \mathcal{T}$ , replace  $\sigma_v^j$  by  $1 - \sigma_v^j$ ;
  - update  $\sigma_v$  according to these changes, and update  $K_v$  to reflect the new sequence length; set the remaining time  $t_\ell := t_\ell - \mathcal{T}$ ;

In words, the evolutionary process defined above assumes that every site of the sequence  $\sigma_u$  of the parent species is, independently from the other sites, subjected to a sequence of evolutionary events that flip its value; these events are distributed according to a Poisson point process of intensity  $\eta_e$  in the time interval  $[0, t_e]$ . However, the site may get deleted and therefore not be inherited by the sequence of the node  $v$ ; this is determined by whether an exponential random variable of rate  $\mu_e$  is smaller than  $t_e$ . While each site of the parental sequence  $\sigma_u$  is subjected to this process, new sites are introduced in the space between existing sites at rate  $\lambda_e$ , and each of these sites follows a similar process for the remaining time.

Given the evolutionary process on a branch of the phylogeny, the evolutionary process on the whole phylogeny is defined as follows.

**Definition 2 (Evolutionary Process).** *Suppose that every site of the sequence  $\sigma_{r(T)}$  at the root of the phylogeny is chosen to be 0 or 1 uniformly at random. Recursively, if  $\sigma_u$  is the sequence at node  $u$  and  $e = (u, v)$  is an edge of the phylogeny, the sequence  $\sigma_v$  at node  $v$  is obtained from the sequence  $\sigma_u$  by an application of the evolutionary process on a branch described by Definition <sup>1</sup>.*

For ease of exposition, we present our proof in the special case when the substitution, insertion and deletion rates are the same on all edges of the phylogeny. We will discuss the more general case in the journal version of the paper.

**Definition 3 (Molecular Clock Assumption).** *Under the molecular clock assumption, there exist  $\eta, \mu$  and  $\lambda$  such that  $\eta_e = \eta, \mu_e = \mu$  and  $\lambda_e = \lambda$ , for all  $e$ .*

*Notation.* In the sequel, we label the leaves of the phylogeny with the positive integers  $1, 2, \dots, n$ , so that  $L = \{1, \dots, n\}$ , and the root  $r(T)$  of the phylogeny with 0.

<sup>4</sup> Clearly, if  $j = 0$ , then  $\sigma_v^j$  is undefined and, if  $j = K_v$ , then  $\sigma_v^{j+1}$  is undefined.

## 1.2 Main Result

*Statement of results.* We begin with a consistency result.

**Theorem 1 (Consistency).** *Assume that  $0 < t_e, \eta_e < +\infty$ , for all  $e \in E$ . Moreover, assume that the indel rates satisfy  $\lambda_e < \mu_e$  for all  $e \in E$ . Under these assumptions, there exists an algorithm solving the phylogenetic reconstruction problem (that is, returning the correct tree) with probability of failure approaching 0 as the sequence length at the root of the tree goes to  $+\infty$ .*

Our main result is the following. For simplicity we work under the symmetric two-state case and assume that the Molecular Clock Assumption holds. We will show in the journal version of the paper that these assumptions are not necessary for our results to hold.

**Theorem 2 (Main Result: Two-State Ultrametric Case).** *Assume there exist constants  $0 < f, g < +\infty$ , independent of  $n$ , such that all branch lengths  $t_e, e \in E$ , satisfy  $f < t_e < g$ . Moreover, assume that  $\eta_e = \eta$ , for all  $e \in E$ , where  $\eta$  is bounded between two constants  $\underline{\eta} > 0$  and  $\bar{\eta} < +\infty$  independent of  $n$ , and that the indel rates satisfy  $\lambda_e = \lambda, \mu_e = \mu$ , for all  $e \in E$ , and  $\lambda < \mu = O(1/\log n)$ . Under the assumptions above, there exists a polynomial-time algorithm solving the phylogenetic reconstruction problem (that is, returning the correct tree) with probability of failure  $O\left(n^{-\beta'}\right)$ , if the root sequence has length  $k_r = \text{poly}_{\beta'}(n)$ .*

The main contribution here is the estimation of an appropriate distance. Once this is done, we simply use a standard reconstruction method, such as Buneman’s method [40].

*Remark 1 (Branch Lengths).* Our assumption that all branch lengths  $t_e, e \in E$ , satisfy  $f < t_e < g$  is standard in the sequence-length requirement literature following the seminal work of [11].

*Remark 2 (Indel Rates).* Under our assumptions about the branch lengths given in Theorem 2 it follows that the evolutionary time from the root of the tree to the leaves is  $\Theta(\log n)$ . This implies that as long as  $\lambda < \mu = O(1/\log n)$ , a constant— independent of  $n$  but potentially arbitrarily small, say 1 in a 100—fraction of the sites of the root sequence “survive” all the way to the leaves of the tree with high probability. Theorem 2 implies that this constant fraction of surviving sites provides sufficient information for the phylogenetic reconstruction problem to be solvable. On the other hand, if the indel rates are significantly higher than  $1/\log n$ , the sequences at the leaves of the tree may experience wild variations in length—a case which appears difficult to analyze.

*Remark 3 (Supercritical Case).* For convenience, our result is stated for the case  $\mu > \lambda$  which is the most relevant in practice. Our algorithm can be extended easily to the cases  $\mu < \lambda$  and  $\mu = \lambda$ . We leave the details to the reader.

*Proof sketch.* As we noted before, unlike the classical setting where the Hamming distance can be analyzed through standard concentration inequalities, proving rigorously that our approach works involves several new technical difficulties. The proof goes through the following steps:

1. **Sequence length and site displacements.** We give bounds on how much sequence lengths vary across the tree, through a moment-generating function argument. Using our bounds on the sequence length process, we bound the worst-case displacements of the sites. Namely we show that, under our assumptions, all sites move by at most  $O(\sqrt{k \log k})$  where  $k$  is the length of the sequence at the root.
2. **Sequence Partitioning.** We divide each sequence in blocks of size roughly  $k^\zeta$  for  $\zeta > 1/2$ . From our bounds on site displacements, it follows that the blocks roughly match across different sequences. In particular, we bound the number of homologous sites between matching blocks with high probability and show that the expected correlation between these blocks is approximately correct.
3. **Expectations.** We compute expectations of block statistics, which involve analyzing a continuous-time Markov process. We use these calculations to define an appropriate additive metric based on correlations between blocks.
4. **Concentration.** Finally, we show that our estimates are concentrated. The concentration argument proceeds by conditioning on the indel process satisfying the high-probability conditions in the previous points.

The crux of our result is the proper estimation of an additive metric. With such an estimation procedure in hand, we can use a standard distance-based approach to recover the phylogeny.

*Organization.* The rest of the paper is organized as follows. The evolutionary distance forming the basis of our approach is presented in Section 2. We describe our full distance estimator in Section 3 and prove its concentration in the same section. All proofs are omitted from this extended abstract. Full proofs and extensions will be described in the journal version of the paper.

## 2 Evolutionary Distances

In this section, we show how to define an appropriate notion of “evolutionary distance” between two species. Although such distances have been widely used in prior phylogenetic work and have been defined for a variety of models [12, 14], to our knowledge our definition is the first that applies to models with indels. We begin by reviewing the standard definition in the indel-free case and then adapt it to the presence of indels. Our estimation procedure is discussed in Section 3. Throughout, we assume  $\mu > \lambda$ .

### 2.1 The Classical Indel-Free Case

Suppose first that  $\lambda_e = \mu_e = 0$  for all  $e$ , that is, *there is no indel*. In that case, the sequence length remains fixed at  $k$  and the alignment problem is trivial. Underlying all distance-based approaches is the following basic definition.

**Definition 4 (Additive Metric).** *A phylogeny is naturally equipped with a so-called additive metric on the leaves  $\mathcal{D} : L \times L \rightarrow (0, +\infty)$  defined as follows  $\forall a, b \in L$ ,  $\mathcal{D}(a, b) = \sum_{e \in P_T(a, b)} \omega_e$ , where  $P_T(a, b)$  is the set of edges on the path between  $a$  and  $b$  in  $T$  and where  $\omega_e$  is a nonnegative function of the parameters on  $e$ —in our case,  $t_e$ ,*

$\eta_e$ ,  $\lambda_e$ , and  $\mu_e$ . For instance, a common choice for  $\omega_e$  would be  $\omega_e = \eta_e t_e$  in which case  $\mathcal{D}(a, b)$  is the expected number of substitutions per site between  $a$  and  $b$ . Often  $\mathcal{D}(a, b)$  is referred to as the “evolutionary distance” between species  $a$  and  $b$ . Additive metrics are characterized by the following four-point condition: for all  $a, b, c, d \in L$ ,  $\mathcal{D}(a, b) + \mathcal{D}(c, d) \leq \max\{\mathcal{D}(a, c) + \mathcal{D}(b, d), \mathcal{D}(a, d) + \mathcal{D}(b, c)\}$ . Moreover, assuming  $\omega_e > 0$  for all  $e \in E$ , it is well-known that there exists a one-to-one correspondence between  $\mathcal{D}$  and  $T$  as a weighed tree with edge weights  $\{\omega_e\}_{e \in E}$ . For more background on tree-based metrics, see [12].

Definition 4 implies that phylogenies can be reconstructed by computing  $\mathcal{D}(a, b)$  for all pairs of leaves  $a, b \in L$ . Assume we seek to estimate the evolutionary distance between species  $a$  and  $b$  using their respective sequences. In a first attempt, one might try the (normalized) Hamming distance between  $\sigma_a = (\sigma_a^1, \dots, \sigma_a^k)$  and  $\sigma_b = (\sigma_b^1, \dots, \sigma_b^k)$ . However, the expected Hamming distance—in other words, the probability of disagreement between a site of  $a$  and  $b$ —does not form an additive metric as defined in Definition 4. Instead, it is well-known that an appropriate estimator is obtained by “correcting” the Hamming distance for “multiple” substitutions. Denoting by  $\hat{\mathcal{H}}(\sigma_a, \sigma_b)$  the Hamming distance between  $\sigma_a$  and  $\sigma_b$ , a Markov chain calculation shows that  $\mathcal{D}(a, b) = -\frac{1}{2} \log(1 - 2\mathbb{E}[\hat{\mathcal{H}}(\sigma_a, \sigma_b)])$ , with the choice  $\omega_e = \eta_e t_e$ . See e.g. [14]. In a distance-based reconstruction procedure, one first estimates  $\mathcal{D}$  with

$$\hat{\mathcal{D}}(a, b) = -\frac{1}{2} \log(1 - 2\hat{\mathcal{H}}(\sigma_a, \sigma_b)), \quad (1)$$

and then applies a standard reconstruction algorithm. The sequence-length requirement of such a method can be derived by using concentration results for  $\hat{\mathcal{H}}$  [11, 15].

## 2.2 Taking Indels into Account

In the presence of indels, the estimator (1) based on the Hamming distance is difficult to apply. One has to first align the sequences, which cannot be done perfectly and causes biases and correlations that are hard to analyze. Alternatively, one could try a different string distance such as edit distance. However, computing the expectation of edit distance under indel models appears difficult.

We use a different approach involving correlations between state frequencies. We will eventually apply the estimator to large sub-blocks of the sequences (see Section 3), but we first describe it for the full sequence for clarity. For a node  $u$ , let  $K_u$  be the (random) length of the sequence at  $u$  and  $Z_u$ , the number of 0’s in the sequence at  $u$ . Then, our distance estimator is  $\hat{\mathcal{D}}(a, b) = (Z_a - \frac{1}{2}K_a)(Z_b - \frac{1}{2}K_b)$ . We now analyze the expectation of this quantity. For  $u \in V$ , we let  $\Delta_u = Z_u - \frac{1}{2}K_u$ , be the deviation of  $Z_u$  from its expected value (conditioned on the sequence length).

*Single channel.* Suppose  $T$  is made of a single edge from the root  $r$  to a leaf  $a$  with parameters  $t, \eta, \lambda, \mu$ . Assume first that the original sequence length is  $k_r = 1$ . Let  $K_a$  be the length of the sequence at  $a$ . Then, by Markov chain calculations [41, Section III.5], it can be shown that the moment-generating function of  $K_a$  is

$$F(s, t) \equiv \mathbb{E}[s^{K_a}] = \frac{\mu(s-1) - e^{(\mu-\lambda)t}(\lambda s - \mu)}{\lambda(s-1) - e^{(\mu-\lambda)t}(\lambda s - \mu)}. \quad (2)$$

By differentiating  $F(s, t)$  we can derive

$$\mathbb{E}[K_a] = e^{-(\mu-\lambda)t}, \tag{3}$$

and

$$\text{Var}[K_a] = \frac{\mu + \lambda}{\mu - \lambda} [e^{-(\mu-\lambda)t} - e^{-2(\mu-\lambda)t}]. \tag{4}$$

Let  $K_a^*$  be the number of “new” sites at  $a$ , that is, excluding the original site if it survived. (We ignore the substitutions for the time being.) The probability that the original site survives is  $e^{-\mu t}$ . Then,  $\mathbb{E}[K_a^*] = \mathbb{E}[K_a - \mathbb{1}\{\text{original site survives}\}] = e^{-(\mu-\lambda)t} - e^{-\mu t}$ , by linearity of expectation.

We now take into account the substitutions. Assume that the original sequence length at  $r$  is a random variable  $K_r$  and that the sequence at  $r$  is i.i.d. uniform. Denote by  $Z_r$  the number of 0’s at  $r$ . The probability that a site in  $r$ , that is still surviving in  $a$ , has flipped its value, that is, has mutated an odd number of times in time  $t$ , is  $p = \sum_{j=0}^{+\infty} e^{-\eta t} \frac{(\eta t)^{2j+1}}{(2j+1)!} = e^{-\eta t} \sinh \eta t = \frac{1 - e^{-2\eta t}}{2}$ . Also, note that a new site created along the path between  $r$  and  $a$  has equal chance of being 0 or 1 *at the end of the path*. Then, we have:

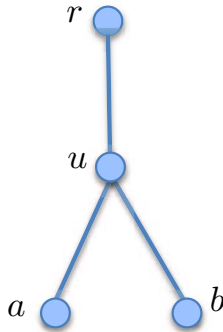
**Lemma 1 (Single Channel: Expected Deviation).**  $\mathbb{E}[\Delta_a | K_r, Z_r] = e^{-(2\eta+\mu)t} \Delta_r$ .

*Fork channel.* Consider now a “fork” tree, that is, a root  $r$  from which emanates a single edge  $e_u = (r, u)$  which in turn branches into two edges  $e_a = (u, a)$  and  $e_b = (u, b)$  (Figure 1). For  $x = a, b, u$ , we denote the parameters of edge  $e_x$  by  $t_x, \lambda_x, \mu_x, \eta_x$ .

**Lemma 2 (Fork Channel: Expected Distance).** *The following holds:*

$$\mathbb{E}[\widehat{\mathcal{D}}(a, b)] = e^{-(2\eta_a + \mu_a)t_a} e^{-(2\eta_b + \mu_b)t_b} e^{-(\mu_u - \lambda_u)t_u} \frac{k_r}{4}.$$

*Molecular clock.* We specialize the previous result to the Molecular Clock Assumption. That is, we assume, for  $x = a, b, u$ , that  $\lambda_x = \lambda, \mu_x = \mu$ , and  $\eta_x = \eta$ . Note that by construction  $t_a = t_b$  (assuming species  $a$  and  $b$  are contemporary). We denote  $t = t_a$  and  $\bar{t} = t_u + t_a$ . Denoting  $\kappa = \frac{k_r e^{-(\mu-\lambda)\bar{t}}}{4}$ , we then get:



**Fig. 1.** The Fork Channel

**Lemma 3 (Molecular Clock: Expected Distance).**  $\mathbb{E} \left[ \widehat{\mathcal{D}}(a, b) \right] = e^{-(4\eta + \mu + \lambda)t} \kappa.$

Letting  $\beta = 4\eta + \mu + \lambda$ , we get that  $-2 \log \mathbb{E}[\kappa^{-1} \widehat{\mathcal{D}}(a, b)] = 2\beta t$ , which is the evolutionary distance between  $a$  and  $b$  with the choice  $\omega_e = \beta t_e$ . Therefore, we define the following estimator  $\widehat{\mathcal{D}}^*(a, b) = -2 \log \kappa^{-1} \widehat{\mathcal{D}}(a, b)$ , where we assume that  $\mu, \lambda, \eta, \kappa$  are known.

### 3 Distance Computation

We now show how to estimate the evolutionary distance between two species by decomposing the sequences into large blocks which serve as roughly independent samples. We use the following notation:  $M_t = e^{-(\mu - \lambda)t}$ ,  $D_t = e^{-\mu t}$ ,  $\delta = \mu - \lambda$ ,  $\phi = \mu + \lambda$ , and  $\Gamma_t = \lambda \delta^{-1} (1 - M_t)$ .

*Remark 4.* Under our main assumptions, the quantities  $M_t$ ,  $D_t$ , and  $\Gamma_t$  are essentially constants, that is,  $O(1)$ . We use this fact throughout this section.

#### 3.1 Concentration of the Indel Process

*Sequence length.* We first show that the sequence length is concentrated. Let  $T$  be single channel consisting of edge  $e = (r, a)$ . Let  $k_r$  be the length at  $r$ .

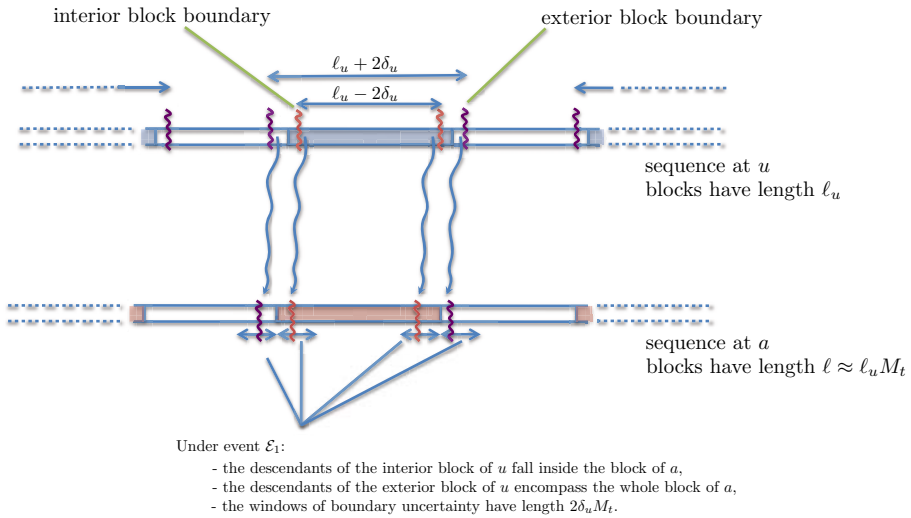
**Lemma 4 (Single Channel: Large Deviations of Sequence Length).** *For all  $\gamma > 0$ , there exists a constant  $c = c(M_t, \Gamma_t; \gamma) > 0$ , such that, for all  $\widehat{k}_r \geq k_r$ , with probability at least  $1 - \widehat{k}_r^{-\gamma}$ , it holds that  $K_a = k_r M_t \pm c \sqrt{\widehat{k}_r \log \widehat{k}_r}$ .*

*Correlated sites.* Now let  $T$  be the fork channel consisting of nodes  $r, u, a$  and  $b$  as in Figure II. Assume that  $a$  and  $b$  are contemporary, call  $t$  the time separating them from  $u$ , and denote by  $S_{ab}$  the number of sites in  $a$  and  $b$  that are jointly surviving from  $u$ . These are the sites that produce correlation between the sequences at  $a$  and  $b$ . All other sites are essentially noise. We bound the large deviations of  $S_{ab}$ .

**Lemma 5 (Fork Channel: Large Deviations of Jointly Surviving Sites).** *Condition on the sequence length at  $u$  being  $k_u$ . Then, for all  $\gamma > 0$ , there exists a constant  $c = c(D_t; \gamma) > 0$ , such that, for all  $\widehat{k}_u \geq k_u$ , with probability at least  $1 - \widehat{k}_u^{-\gamma}$ , it holds that  $S_{ab} = k_u D_t^2 \pm c \sqrt{\widehat{k}_u \log \widehat{k}_u}$ .*

#### 3.2 Sequence Partitioning

From Lemma 4, it follows that the sites of the root sequence (or of an internal sequence) remain fairly close to their expected position at the leaves. We take advantage of this fact by dividing each sequence into blocks of size asymptotically larger than the typical displacement implied by Lemma 4. As a result, matching blocks in different sequences share a significant fraction of sites. Moreover, distinct blocks are roughly independent.



**Fig. 2.** Under the event  $\mathcal{E}_1$  the descendants of the interior blocks of  $\sigma_u$  fall inside the corresponding blocks of  $\sigma_a$ ; the descendants of the exterior blocks of  $\sigma_u$  contain all surviving sites inside the corresponding blocks of  $\sigma_a$ ; the windows of uncertainty have length  $2M_t\delta_u$

We estimate the evolutionary distance between two leaves by comparing the site frequencies in matching blocks. This requires some care as we show next.

Consider the fork channel. We seek to estimate the evolutionary distance  $\widehat{D}(a, b)$  between  $a$  and  $b$  (normalized by the sequence length at  $u$ ). See Figure 2 for an illustration of the partitioning of sequences that is described next.

*Partitioning the leaf sequences.* Let  $k_0$  be some *deterministic* length (to be determined), and consider the first  $k_0$  sites in the sequences  $\sigma_a$  and  $\sigma_b$  at the nodes  $a$  and  $b$  respectively. If the sequence at  $a$  or  $b$  has length smaller than  $k_0$ , we declare that our distance estimate  $\widetilde{D}(a, b)$  (see below) is  $+\infty$ .

We divide the leaf sequences into  $L$  blocks of length  $\ell$  where  $\ell = \lceil k_0^\zeta \rceil$ , for some  $\frac{1}{2} < \zeta < 1$  to be determined later, and  $L = \lfloor k_0/\ell \rfloor$ . We let  $k'_0 = \ell L$ . For all  $i = 1, \dots, L$ , we define the  $i$ -th *block*  $\sigma_{a,i}$  of  $a$  to be the subsequence of  $\sigma_a$  ranging from position  $(i - 1)\ell + 1$  to position  $i\ell$ . We let  $Z_{a,i}$  be the number of zeros inside  $\sigma_{a,i}$  and define the *block deviations*  $\Delta_{a,i} = Z_{a,i} - \frac{\ell}{2}$ , for all  $i = 1, \dots, L$ . And similarly for the sequence at  $b$ .

Using the above notation we define our distance estimator next. Assume that  $L$  is even. Otherwise, we can just drop the last block in the above partition. Our estimator is the following:  $\widetilde{D}(a, b) = \frac{2}{L} \sum_{j=0}^{L/2-1} \Delta_{a,2j+1} \Delta_{b,2j+1}$ . Notice that in our summation above we skipped every other block in our sequence partition to avoid overlapping sites and hence decrease potential correlations between the terms in the estimator. In the rest of this section, we analyze the properties of  $\widetilde{D}(a, b)$ . To do this it is helpful to consider the sequence at  $u$  and the events that happened in the channels defined by the edges  $(u, a)$  and  $(u, b)$ .



*Partitioning the ancestral sequence.* Let us choose  $\ell_u$  to be the largest integer satisfying

$$\ell_u M_t \leq \ell. \tag{5}$$

Suppose that the sequence  $\sigma_u$  at node  $u$  is not shorter than  $k'_u = (L - 1)\ell_u$ , and define the  $i$ -th *ancestral block*  $\sigma_{u,i}$  of  $u$  to be the subsequence of  $\sigma_u$  ranging from position  $(i - 1)\ell_u + 1$  to position  $i\ell_u$ , for all  $i \leq L - 1$ . Given Lemma 4, the choice of  $\ell_u$  in (5) is such that the blocks of  $u$  and the corresponding blocks at  $a$  and  $b$  roughly align.

In order to use the expected evolutionary distance as computed in Lemma 3, we define an “interior” ancestral block which is guaranteed with high probability to remain entirely “inside” the corresponding leaf block. Let  $\delta_u = \lceil L + \frac{c}{M_t} \sqrt{k'_u \log k'_u} \rceil$ , where  $c$  is an appropriate constant. (The  $L = o(\sqrt{k_0})$  in  $\delta_u$  is needed only when (5) is a strict inequality.) We define the  $i$ -th (*ancestral*) *interior block*  $\sigma'_{u,i}$  of  $u$  to be the subsequence of  $\sigma_{u,i}$  ranging from position  $(i - 1)\ell_u + \delta_u$  of  $\sigma_u$  to position  $i\ell_u - \delta_u$ . Notice that  $\delta_u \sim \sqrt{k_0 \log k_0}$ , while  $\ell_u \sim k_0^\zeta$ . Therefore, for  $k_0 > k_0^*$ , where  $k_0^* = k_0^*(\mu, \lambda, t, \gamma) > 0$  is sufficiently large,  $(i - 1)\ell_u + \delta_u \ll i\ell_u - \delta_u$  so that the sequence  $\sigma'_{u,i}$  is well-defined.

Also, for all  $i = 1, \dots, L - 1$ , we define  $x'_{a,i}, y'_{a,i}$  to be the position of the leftmost, respectively rightmost, site in the sequence  $\sigma_a$  descending from the site at position  $(i - 1)\ell_u + \delta_u$ , respectively  $i\ell_u - \delta_u$  of  $\sigma_u$ . Similarly we define  $x'_{b,i}$  and  $y'_{b,i}$ . Given this notation, we define the following “good” event

$$\mathcal{E}'_1 = \{ \forall i \leq L - 1 : (i - 1)\ell < x'_{a,i}, x'_{b,i} < (i - 1)\ell + 2M_t\delta_u, \\ i\ell - 2M_t\delta_u < y'_{a,i}, y'_{b,i} < i\ell \}. \tag{6}$$

Intuitively, when the event  $\mathcal{E}'_1$  holds, all descendants of the interior block  $\sigma'_{u,i}$  are located inside the blocks  $\sigma_{a,i}$  and  $\sigma_{b,i}$  respectively (and they do not shrink much).

To argue about block independence, we also define the *exterior block*  $\sigma''_{u,i}$  of  $u$  to be the subsequence of  $\sigma_{u,i}$  ranging from position  $(i - 1)\ell_u - \delta_u$  of  $\sigma_u$  to position  $i\ell_u + \delta_u$  with corresponding positions  $x''_{a,i}, y''_{a,i}, x''_{b,i}$  and  $y''_{b,i}$  and good event  $\mathcal{E}''_1$  defined similarly as above.

We show that the event  $\mathcal{E}_1 = \mathcal{E}'_1 \cup \mathcal{E}''_1$  holds with high probability, conditioned on the sequence length  $K_u$  at  $u$  being at least  $k'_u$ . Figure 2 shows the structure of the indel process in the case that the event  $\mathcal{E}_1$  holds.

**Lemma 6 (Interior/Exterior Block Is Inside/Outside Leaf Block).** *Conditioned on the event  $\{K_u \geq k'_u\}$ , we have  $\mathbb{P}[\mathcal{E}_1] \geq 1 - 16L \left(\frac{1}{k'_u}\right)^\gamma$ .*

*Block correlation.* Let  $S_{ab,i}$  be the number of common sites in the blocks  $\sigma_{a,i}$  and  $\sigma_{b,i}$  that are jointly surviving from  $u$ . Similarly we define  $S'_{ab,i}$  and  $S''_{ab,i}$  where, for  $\xi = a, b$ ,  $\sigma'_{\xi,i}$  ( $\sigma''_{\xi,i}$ ) denotes the subsequence of  $\sigma_\xi$  ranging from position  $x'_{\xi,i}$  ( $x''_{\xi,i}$ ) to position  $y'_{\xi,i}$  ( $y''_{\xi,i}$ ). We define a good event for  $S_{ab,i}$  as

$$\mathcal{E}_2 = \{ \forall i \leq L - 1 : \ell_u D_t^2 - 3M_t\delta_u \leq S_{ab,i} \leq \ell_u D_t^2 + 3M_t\delta_u \}.$$

**Lemma 7 (Jointly Surviving Sites in Blocks).** *Conditioned on the event  $\{K_u \geq k'_u\}$ , we have  $\mathbb{P}[\mathcal{E}_2] \geq 1 - 18L \left(\frac{1}{k'_u}\right)^\gamma$ .*



### 3.3 Estimation Guarantees

We are now ready to analyze the behavior of our estimate  $\tilde{\mathcal{D}}(a, b)$ . In this subsection we compute the expectation and variance of  $\tilde{\mathcal{D}}(a, b)$ . We denote by  $\mathcal{I}$  a realization of the indel process (but not of the substitution process) on the paths between  $u$  and  $a, b$ . We denote by  $\mathcal{E}$  the event such that  $\{K_u \geq k'_u\}$ ,  $\mathcal{E}_1$ , and  $\mathcal{E}_2$  are satisfied. Suppose that  $k_0 > k_0^*$ .

**Lemma 8 (Block Independence).** *Conditioning on  $\mathcal{I}$  and  $\mathcal{E}$ ,  $\{\Delta_{a,2j+1}\Delta_{b,2j+1}\}_{j=1}^{L/2-1}$  are mutually independent.*

**Lemma 9 (Expected Correlation under Good Event).** *We have*

$$\mathbb{E}[\Delta_{a,i}\Delta_{b,i} \mid \mathcal{I}, \mathcal{E}] = \frac{1}{4}e^{-4\eta t}e^{-2\mu t}\ell_u \pm O\left(\sqrt{k_0 \log k_0}\right).$$

**Lemma 10 (Variance under Good Event).** *We have  $\text{Var}[\Delta_{a,i}\Delta_{b,i} \mid \mathcal{I}, \mathcal{E}] \leq \frac{3}{16}\ell^2$ .*

**Lemma 11 (Distance Estimate).** *We have*

$$\mathbb{E}\left[\tilde{\mathcal{D}}(a, b) \mid \mathcal{I}, \mathcal{E}\right] = \frac{1}{4}e^{-(4\eta+\mu+\lambda)t}\ell \pm O\left(\sqrt{k_0 \log k_0}\right),$$

and  $\text{Var}\left[\tilde{\mathcal{D}}(a, b) \mid \mathcal{I}, \mathcal{E}\right] \leq \frac{3}{8}\frac{1}{[k_0^{1-\zeta}]} \ell^2$ . In particular, for  $\zeta > 1/2$  small enough

$$\text{STD}\left[\tilde{\mathcal{D}}(a, b) \mid \mathcal{I}, \mathcal{E}\right] = O\left(k_0^{\frac{3\zeta-1}{2}}\right) = o(\sqrt{k_0}).$$

### 3.4 Concentration

We now show that our distance estimate is concentrated. For notational convenience, we denote by  $\mathbb{P}_u^*$  the probability measure induced by conditioning on the event  $\{K_u \geq k'_u\}$ . Recall that the event  $\mathcal{E}$  is contained in  $\{K_u \geq k'_u\}$ .

**Lemma 12 (Concentration of Distance Estimate).** *Let  $\alpha > 0$  be such that  $\zeta - \alpha > 1/2$ , and  $\beta = 1 - \zeta - 2\alpha > 0$ , for  $\zeta > 1/2$  small enough. Then for  $k_0$  large enough*

$$\mathbb{P}_u^* \left[ \left| \frac{4}{\ell} \tilde{\mathcal{D}}(a, b) - e^{-(4\eta+\mu+\lambda)t} \right| > \frac{1}{k_0^\alpha} \right] \leq O\left(\frac{1}{k_0^\beta}\right).$$

The proofs of Theorems 1 and 2 follow using the standard Buneman algorithm [40].

## References

1. Thorne, J.L., Kishino, H., Felsenstein, J.: An evolutionary model for maximum likelihood alignment of dna sequences. *Journal of Molecular Evolution* 33(2), 114–124 (1991)
2. Thorne, J.L., Kishino, H., Felsenstein, J.: Inching toward reality: An improved likelihood model of sequence evolution. *Journal of Molecular Evolution* 34(1), 3–16 (1992)

3. Loytynoja, A., Goldman, N.: Phylogeny-Aware Gap Placement Prevents Errors in Sequence Alignment and Evolutionary Analysis. *Science* 320(5883), 1632–1635 (2008)
4. Wong, K.M., Suchard, M.A., Huelsenbeck, J.P.: Alignment Uncertainty and Genomic Analysis. *Science* 319(5862), 473–476 (2008)
5. Metzler, D.: Statistical alignment based on fragment insertion and deletion models. *Bioinformatics* 19(4), 490–499 (2003)
6. Miklos, I., Lunter, G.A., Holmes, I.: A "Long Indel" Model For Evolutionary Sequence Alignment. *Mol. Biol. Evol.* 21(3), 529–540 (2004)
7. Suchard, M.A., Redelings, B.D.: BALi-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics* 22(16), 2047–2048 (2006)
8. Rivas, E., Eddy, S.R.: Probabilistic phylogenetic inference with insertions and deletions. *PLoS Comput. Biol.* 4, e1000172 (2008)
9. Liu, K., Raghavan, S., Nelesen, S., Linder, C.R., Warnow, T.: Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees. *Science* 324(5934), 1561–1564 (2009)
10. Felsenstein, J.: Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Biol.*, 401–410 (1978)
11. Erdős, P.L., Steel, M.A., Székely, L.A., Warnow, T.A.: A few logs suffice to build (almost) all trees (part 1). *Random Struct. Algor.* 14(2), 153–184 (1999)
12. Semple, C., Steel, M.: *Phylogenetics. Mathematics and its Applications series, vol. 22.* Oxford University Press, Oxford (2003)
13. Graur, D., Li, W.-H.: *Fundamentals of Molecular Evolution*, 2nd edn. Sinauer Associates, Inc., Sunderland (1999)
14. Felsenstein, J.: *Inferring Phylogenies.* Sinauer, New York (2004)
15. Atteson, K.: The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica* 25(2-3), 251–278 (1999)
16. Erdős, P.L., Steel, M.A., Székely, L.A., Warnow, T.A.: A few logs suffice to build (almost) all trees (part 2). *Theor. Comput. Sci.* 221, 77–118 (1999)
17. Huson, D.H., Nettles, S.H., Warnow, T.J.: Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *J. Comput. Biol.* 6(3–4) (1999)
18. Steel, M.A., Székely, L.A.: Inverting random functions. *Ann. Comb.* 3(1), 103–113 (1999); *Combinatorics and biology* (Los Alamos, NM, 1998)
19. Csurös, M., Kao, M.Y.: Provably fast and accurate recovery of evolutionary trees through harmonic greedy triplets. *SIAM Journal on Computing* 31(1), 306–322 (2001)
20. Csurös, M.: Fast recovery of evolutionary trees with thousands of nodes. *J. Comput. Biol.* 9(2), 277–297 (2002)
21. Steel, M.A., Székely, L.A.: Inverting random functions. II. Explicit bounds for discrete maximum likelihood estimation, with applications. *SIAM J. Discrete Math.* 15(4), 562–575 (2002) (electronic)
22. King, V., Zhang, L., Zhou, Y.: On the complexity of distance-based evolutionary tree reconstruction. In: *SODA 2003: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 444–453. Society for Industrial and Applied Mathematics, Philadelphia (2003)
23. Mossel, E., Roch, S.: Learning nonsingular phylogenies and hidden Markov models. *Ann. Appl. Probab.* 16(2), 583–614 (2006)
24. Daskalakis, C., Mossel, E., Roch, S.: Optimal phylogenetic reconstruction. In: *STOC 2006: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pp. 159–168. ACM Press, New York (2006)
25. Lacey, M.R., Chang, J.T.: A signal-to-noise analysis of phylogeny estimation by neighbor-joining: insufficiency of polynomial length sequences. *Math. Biosci.* 199(2), 188–215 (2006)

26. Daskalakis, C., Hill, C., Jaffe, A., Mihaescu, R., Mossel, E., Rao, S.: Maximal accurate forests from distance matrices. In: Apostolico, A., Guerra, C., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2006. LNCS (LNBI), vol. 3909, pp. 281–295. Springer, Heidelberg (2006)
27. Mossel, E.: Distorted metrics on trees and phylogenetic forests. *IEEE/ACM Trans. Comput. Bio. Bioinform.* 4(1), 108–116 (2007)
28. Gronau, I., Moran, S., Snir, S.: Fast and reliable reconstruction of phylogenetic trees with very short edges. In: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 379–388. Society for Industrial and Applied Mathematics, Philadelphia (2008)
29. Roch, S.: Sequence-length requirement for distance-based phylogeny reconstruction: Breaking the polynomial barrier. In: FOCS, pp. 729–738 (2008)
30. Daskalakis, C., Mossel, E., Roch, S.: Phylogenies without branch bounds: Contracting the short, pruning the deep. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 451–465. Springer, Heidelberg (2009)
31. Wang, L., Jiang, T.: On the complexity of multiple sequence alignment. *Journal of Computational Biology* 1(4), 337–348 (1994)
32. Elias, I.: Settling the intractability of multiple alignment. *Journal of Computational Biology* 13(7), 1323–1339 (2006) PMID: 17037961
33. Higgins, D.G., Sharp, P.M.: Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73(1), 237–244 (1988)
34. Katoh, K., Misawa, K., Kuma, K.: MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucl. Acids Res.* 30(14), 3059–3066 (2002)
35. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.* 32(5), 1792–1797 (2004)
36. Thatte, B.D.: Invertibility of the TKF model of sequence evolution. *Math. Biosci.* 200(1), 58–75 (2006)
37. Andoni, A., Daskalakis, C., Hassidim, A., Roch, S.: Trace reconstruction on a tree (2009) (Preprint)
38. Hohl, M., Ragan, M.A.: Is Multiple-Sequence Alignment Required for Accurate Inference of Phylogeny? *Syst. Biol.* 56(2), 206–221 (2007)
39. Karlin, S., Taylor, H.M.: A second course in stochastic processes, p. 542. Academic Press Inc.[Harcourt Brace Jovanovich Publishers], New York (1981)
40. Buneman, P.: The recovery of trees from measures of dissimilarity. In: *Mathematics in the Archaeological and Historical Sciences*, pp. 187–395. Edinburgh University Press, Edinburgh (1971)
41. Athreya, K.B., Ney, P.E.: *Branching processes*. Springer, New York (1972); *Die Grundlehren der mathematischen Wissenschaften, Band 196*

# Inference of Isoforms from Short Sequence Reads (Extended Abstract)

Jianxing Feng<sup>1</sup>, Wei Li<sup>2</sup>, and Tao Jiang<sup>3</sup>

<sup>1</sup> State Key Laboratory on Intelligent Technology and Systems,  
Tsinghua National Laboratory for Information Science and Technology,  
Department of Computer Science, Tsinghua Univ., Beijing, China  
fengjx06@mails.tsinghua.edu.cn

<sup>2</sup> Department of Computer Science, Univ. of California, Riverside, CA  
liw@cs.ucr.edu

<sup>3</sup> Department of Computer Science, Univ. of California, Riverside, CA, and Tsinghua  
Univ., Beijing, China  
jiang@cs.ucr.edu

**Abstract.** Due to alternative splicing events in eukaryotic species, the identification of mRNA isoforms (or splicing variants) is a difficult problem. Traditional experimental methods for this purpose are time consuming and cost ineffective. The emerging RNA-Seq technology provides a possible effective method to address this problem. Although the advantages of RNA-Seq over traditional methods in transcriptome analysis have been confirmed by many studies, the inference of isoforms from millions of short sequence reads (*e.g.*, Illumina/Solexa reads) has remained computationally challenging. In this work, we propose a method to calculate the expression levels of isoforms and infer isoforms from short RNA-Seq reads using exon-intron boundary, transcription start site (TSS) and poly-A site (PAS) information. We first formulate the relationship among exons, isoforms, and single-end reads as a convex quadratic program, and then use an efficient algorithm (called IsoInfer) to search for isoforms. IsoInfer can calculate the expression levels of isoforms accurately if all the isoforms are known and infer novel isoforms from scratch. Our experimental tests on known mouse isoforms with both simulated expression levels and reads demonstrate that IsoInfer is able to calculate the expression levels of isoforms with an accuracy comparable to the state-of-the-art statistical method and a 60 times faster speed. Moreover, our tests on both simulated and real reads show that it achieves a good precision and sensitivity in inferring isoforms when given accurate exon-intron boundary, TSS and PAS information, especially for isoforms whose expression levels are significantly high.

## 1 Introduction

Transcriptome study (or transcriptomics) aims to discover all the transcripts and their quantities in a cell or an organism under different external environmental conditions. A large amount of work has been devoted to transcriptomics, which

includes the international projects EST [11,2], FANTOM [3], and ENCODE [4,5]. Many technologies have been introduced in recent years including array-based experimental methods such as tiling arrays [6], exon arrays [7], and exon-junction arrays [8,9], and tag-based approaches such as MPSS [10,11], SAGE [12,13], CAGE [14,15], PMAGE [16], and GIS [17]. However, due to various constraints intrinsic to these technologies, the speed of advance in transcriptomics is far from being satisfactory, especially on eukaryotic species because of widespread alternative splicing events.

Applying next generation sequencing technologies to transcriptomes, the recently developed RNA-Seq technology is quickly becoming an important tool in functional genomics and transcriptomics. It can be used to identify all genes and exons and their boundaries [18,19] and to study gene functions and perform transcriptome analysis [20]. For example, based on an unannotated genomic sequence and millions of short reads from RNA-Seq, [21] developed a general method for the discovery of a complete transcriptome, including the identification of coding regions, ends of transcripts, splice junctions, splice site variations, *etc.* Their application of the method to *S.cerevisiae* (yeast) showed a high degree of agreement with the existing knowledge of the yeast transcriptome. Besides yeast [22,18], RNA-Seq has been applied to the transcriptome analysis of mouse [23,24] and human [25,26]. These results demonstrate that RNA-Seq is a powerful quantitative method to sample a transcriptome deeply at an unprecedented resolution. Moreover, DNA sequencing technologies are under fast development. Some of them now could provide long reads, paired-end reads, DNA-strand-sequencing of mRNA transcripts, *etc.* See [27] for a comprehensive analysis of the advantages of RNA-Seq over traditional methods in genome-wide transcriptome analysis, and the challenges faced by this technology.

Very recently, several methods have been proposed to characterize the expression level of each transcript [28,29] using RNA-Seq data. In [28], the authors showed that short (single-end or paired-end) read sequences cannot theoretically guarantee a unique solution to the *transcriptome reconstruction* problem (*i.e.*, the reconstruction of all expressed isoforms and their expression levels) in general even if the reads are sampled perfectly according to the length of each transcript (without random distortions and noise). However, under the same assumption, the authors also showed that paired-end reads could help reconstruct the transcripts uniquely and determine their expression levels for most of the currently known isoforms of human, and single-end reads could allow us to determine the expression levels correctly if all the isoforms are known. However, these results are mostly of theoretical interest because sequence read data are random in nature and may contain noise in practice. [29] presented a more practical way to estimate the expression levels of known isoforms. The method uses maximum likelihood estimation followed by importance sampling from the posterior distribution.

The availability of all the isoforms is the basis of the accurate estimation of isoform expression levels [29], which could be used to infer all splicing variants quantitatively and qualitatively. The variations in isoform expression levels and

splicing are important for many studies, *e.g.*, the study of diseases [30,31] and drug development [32]. A lot of effort has been devoted to the identification of transcripts/isoforms from the more traditional EST, cDNA data. Instead of a comprehensive review, we will just name a few results below. To enumerate all possible isoforms, a core ingredient is the *splicing graph* [33,34]. A predetermined parameter “dimension” decides how many transcripts are compared simultaneously. The parameter is usually fixed to two, but recently, [34] extended the method to arbitrary dimensions. There are several methods that assemble transcripts from EST data using the splicing graph and its variations [35,36]. Newly proposed experimental methods in [37,38] could be used to identify new isoforms. However, it is still unclear whether these methods can be applied in a large scale.

RNA-Seq has shown great success in transcriptome analysis, but it has not been used to infer isoforms. Although it is straightforward to infer the existence of novel isoforms from RNA-Seq data that exhibit novel transcribed regions [24,6], it is not so obvious how to use RNA-Seq data to infer the existence of novel isoforms in known transcribed regions, because the observed reads could be sampled from either known or unknown isoforms. The problem has remained challenging for two reasons. The first is that RNA-Seq reads are usually very short. The second is due to the randomness and biases of the reads sampled from all the transcripts. In fact, to our best knowledge, there has been no published work to computationally infer isoforms from (realistic) short RNA-Seq reads.

Due to the high combinatorial complexity of isoforms of genes with a (moderately) large number of exons, the inference of isoforms from short reads (and other available biological information) should be realistically divided into two sub-problems. The first is to discover all the exon-intron boundaries as well as the transcription start site (TSS) and poly-A site (PAS) of each isoform. As mentioned above, there are several effective methods for detecting exon-intron boundaries from RNA-Seq read data [18,19]. The identification of TSS’s and PAS’s is an indispensable part of many large genomics projects [3,4]. The technology of GIS-PET (Gene Identification Signature Paired-End Tags) can also be used to identify TSS-PAS pairs [17,39]. The second sub-problem is to find combinations of exons that can properly explain the RNA-Seq data, given the exon-intron boundary and TSS-PAS pair information.

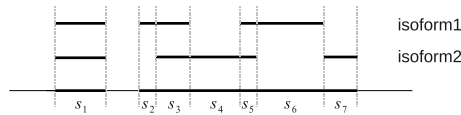
In this paper, we are concerned with the second sub-problem in isoform inference. Assuming that the exon-intron boundary and TSS-PAS pair information is given, we propose a method (called IsoInfer) to infer isoforms from short RNA-Seq reads (*e.g.*, Illumina/Solexa data). Although our method works for single-end data and data with both single-end and paired-end reads, we will use single-end reads as the primary source of data and paired-end reads as a secondary data which can be used to filter out false positives. We formulate the relationship among exons, isoforms, and single-end reads as a convex quadratic program, and design an efficient algorithm to search for isoforms. Our method can calculate the expression levels of isoforms accurately if all the isoforms are known. To demonstrate this, we have compared IsoInfer with the simple counting method in [40,41] and the method in [29] on simulated expression levels and

reads, and found that our method is much more accurate than the simple counting method and has a comparable accuracy as the method in [29] but is 60 times faster. Most importantly, IsoInfer can infer isoforms from scratch when they are sufficiently expressed, by trying to find a minimum set of isoforms to explain the read data. Our experimental tests on both simulated and real reads show that it is possible to infer the precise combination of exons in a sufficiently expressed isoform from RNA-Seq short read data with a reasonably good accuracy, when accurate exon-intron boundary and TSS-PAS pair information is provided. To our best knowledge, this is the first computational method to infer isoforms from short RNA-Seq reads. Due to the page limit, some proofs and tables are omitted in this extended abstract but can be found in the full paper [42].

## 2 Methods

### 2.1 Assumptions and Terminology

Traditionally, only five types of alternating splicing (AS) events have been proposed, including exon skipping, mutually exclusive exons, intron retention, alternative donor and acceptor sites [43]. However, these events are not adequate to describe complex AS events as more experimental knowledge has become available [44]. In this work, we describe isoforms or AS events in a much general way, which is referred to as a “bit matrix” in [44].



**Fig. 1.** Expressed segments. Every exon-intron boundary introduces a boundary of some segment. Every expressed segment is a part of an exon.

The exon-intron boundaries of a gene divide the gene into disjoint *segments*, as shown in Figure 1. A segment is *expressed* if it has mapped reads. Thus, every expressed isoform consists of a subset of expressed segments. Two segments are adjacent if they are adjacent in the reference genome (*i.e.*, they share a common boundary). For example, in Figure 1,  $s_2$  and  $s_3$  are adjacent but  $s_1$  and  $s_2$  are not. Any two segments may form a *segment junction* which is not necessarily an exon junction in the traditional sense. For example,  $s_2$  and  $s_3$  form a segment junction, which is not an exon junction. In the following, “junction” refers to “segment junction” unless otherwise stated.

As stated in the introduction, we first assume that exon-intron boundaries are known. Our second assumption is that the short reads are uniformly randomly sampled from all the expressed isoforms (*i.e.*, mRNA transcripts). We have to further assume that the short reads have been mapped to the referenced genome. The mapping of RNA-Seq reads can be done by many recent tools, *e.g.*, Bowtie [45], Maq [46], SOAP [47], RNA-MATE [48] and mrFAST [49]. The mapping of multi-reads (*i.e.*, reads that match several locations of the reference genome) is addressed



in [24,50]. We will use Bowtie in our work due to its efficiency and accuracy. The last assumption concerns paired-end reads, which will be stated in section 2.3.

## 2.2 Quadratic Programming Formulation

$\mathcal{G}$  denotes the set of all the genes. Each  $g$  gene defines a set of expressed segments  $S_g = \{s_1, s_2, \dots, s_{|S_g|}\}$  (given exon-intron boundaries), where the expressed segments are sorted according to their positions in the reference genome. The junctions on this gene are all the pairs of expressed segments  $(s_i, s_j), 1 \leq i < j \leq |S_g|$ . The length of segment  $s_i$  is  $l_i$ . Denote the set of all known isoforms of this gene as  $F_g$ . Each isoform  $f \in F_g$  consists of a subset of expressed segments. The expression level (*i.e.*, the number of reads per base) of isoform  $f$  is denoted by  $x_f$ . The sum of the length of all transcripts, weighted by their expression levels, over all genes, is  $L_0 = C \cdot \sum_{g \in \mathcal{G}} \sum_{s \in f, f \in F_g} l_s x_f$ , for some constant  $C$  that defines the linear relationship between the expression level and the number of transcripts corresponding to an isoform.  $C$  can be inferred from data as shown in [24].

From now on, we will consider a fixed gene  $g$  and omit the subscript  $g$  when there is no ambiguity. Let  $M$  be the total number of single-end reads mapped to the reference genome and  $d_i$  the number of reads falling into expressed segment  $s_i$ . Under the uniform sampling assumption,  $d_i$  is the observed value of a random variable (denoted as  $r_i$ ) that follows the binomial distribution  $B(M, p_i)$ , where  $p_i = Cy_i l_i / L_0$  and  $y_i = \sum_{s_i \in f} x_f$ . Because  $M$  is usually very large,  $p_i$  is very small and  $Mp_i$  is sufficiently large in most cases, the binomial distribution can be approximated by a normal distribution  $N(\mu_i, \sigma_i^2)$ , with  $\mu_i = Mp_i, \sigma_i^2 = Mp_i(1 - p_i) \approx Mp_i = \mu_i$ , similar to the approximation in [29]. Therefore, the random variables  $\frac{r_i - \mu_i}{\sigma_i}$ , for every expressed segment  $s_i$ , follow the same distribution approximately. Define  $\epsilon_i = |r_i - \mu_i|$ . Then, the variable  $\frac{\epsilon_i}{\sigma_i}$  also follows the same distribution approximately for every  $s_i$ .

Let  $L_1$  denote the length of a single-end read. In order to map reads to junctions, we will also think of each junction  $(s_i, s_j)$  as a segment of length  $2L_1 - 2$ , consisting of the last  $L_1 - 1$  bases of  $s_i$  and the first  $L_1 - 1$  bases of  $s_j$ . Denote the set of the junctions as  $J = \{s_{|S|+1}, s_{|S|+2}, \dots, s_{|S|+|J|}\}$ . The relationship among the expressed segments of gene  $g$ , its expressed isoforms, and the single-end reads mapped to each expressed segment and junction can be captured by the following quadratic program (QP):

$$\begin{aligned} \min \quad & z = \sum_{s_i \in S \cup J} \left(\frac{\epsilon_i}{\sigma_i}\right)^2 \\ \text{s.t.} \quad & \sum_{s_i \in f} x_f l_i + \epsilon_i = d_i, \quad s_i \in S \cup J \\ & x_f \geq 0, \quad f \in F \end{aligned}$$

where  $\sigma_i$  is the standard deviation in the normal distribution  $N(\mu_i, \sigma_i^2)$  and will be empirically estimated from  $d_i$ .

Note that if each  $r_i$  follows the normal distribution strictly, then the random variables  $\frac{\epsilon_i}{\sigma_i}$  is i.i.d. and thus the solution of the above QP would correspond to the maximum likelihood estimation of the  $x_f$ 's if each  $\sigma_i$  is fixed [51], and the objective function  $z$  is a random variable obeying the  $\chi^2$  distribution with freedom



$|S| + |J|$ . This QP can be easily shown to be a convex QP by a simple transformation and solved in polynomial time by a public program QuadProg++ which implements the dual method of Goldfarb and Idnani [52] for convex quadratic programming. Since  $\sigma_i$  is unknown, we substitute  $\sqrt{d_i}$  for  $\sigma_i$  as an approximation. Let QPsolver denote the above algorithm for solving the convex QP program. Given  $S, F$ , and  $d_i$ 's, QPsolver returns the values of  $x_i$ 's and  $z$ .

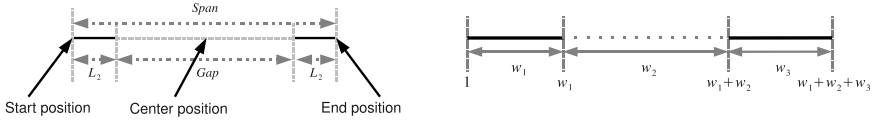
When the isoforms in  $F$  are given, minimizing the objective function means to find a combination of the expression level ( $x_f$ ) of each isoform in  $F$  such that the observed values ( $d_i$ 's) can be explained the best. In this case, the value of the objective function serves as an indicator of whether the isoforms in  $F$  can explain the observed data. More specifically,  $p\text{-value}(z)$  denotes the probability of  $P(Z \geq z)$ , where  $Z$  is a random variable following the  $\chi^2$  distribution with freedom  $|S| + |J|$ . We empirically choose a cutoff of 0.05. If  $p\text{-value}(z)$  is less than 0.05 we conclude that  $F$  cannot explain  $d$ .

### 2.3 Paired-End Reads

Figure 2(left) illustrates some concepts concerning paired-end reads. A paired-end read consists of a pair of short (single-end) reads separated by a *gap*. The figure also defines the *read length*, *span*, *start position*, *center position* and *end position* of a paired-end read. If the span of a paired-end read is a random variable following some probability distribution  $h(x)$ , then three possible strategies for generating paired-end reads will be considered in this paper.

- Strategy (a): The start position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution  $h(x)$ . If the end position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the end position of this read is at the end of the isoform.
- Strategy (b): The center position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution  $h(x)$ . This strategy has been adopted in [53]. Again, if the start (or end) position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the start (or end, respectively) position of this read is at the start (or end, respectively) of the isoform.
- Strategy (c): The end position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution  $h(x)$ . If the start position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the start position of this read is at the start of the isoform.

Let  $w_1, w_2, w_3$  be the lengths of three consecutive intervals on an isoform as shown in Figure 2(right). When any of the strategies (a-c) is applied to generate a certain number of paired-end reads, the following Theorem 1 gives a non-trivial upper bound on the probability of not observing any reads with start positions in the first interval and end positions in the third interval.



**Fig. 2.** Left: A paired-end read consisting of two short reads of length  $L_2$  that are separated by a gap. Right: Three consecutive intervals on an isoform.

**Theorem 1.** *Suppose that the expression level of this isoform is  $\alpha$  RPKM (i.e., reads per kilobase of exon model per million mapped reads [24]), and the span of each paired-end read follows some distribution  $h(x)$ . If  $M$  paired-end reads are generated by any of the strategies (a-c), the probability that there are no paired-end reads that have start positions in the first interval and end positions in the third interval is upper bounded by*

$$P_{M,h,\alpha}(w_1, w_2, w_3) = (1 - P_0)^M \approx e^{-MP_0}$$

where  $P_0 = 10^{-9}\alpha \sum_{0 \leq i < w_1} \int_{l(i)}^{u(i)} h(x)dx$ ,  $l(i) = w_1 - i + w_2$ , and  $u(i) = w_1 - i + w_2 + w_3$ .

### 2.4 Valid Isoforms

For a gene with expressed segments  $S = \{s_1, s_2, \dots, s_{|S|}\}$ , an isoform  $f$  of this gene can be expressed as a binary vector with length  $|S|$ . The  $i$ th element  $f[i]$  of  $f$  is 1 if and only if expressed segment  $s_i$  is contained in  $f$ . Denote the set of all possible binary vectors with  $n$  elements as  $B(n)$ . Similarly, a single-end or paired-end short read that is mapped to a subset  $S' \subseteq S$  of expressed segments can be represented as a binary vector  $r \in B(|E|)$  such that  $r[i] = 1$  if and only if  $s_i \in E'$ . A subset  $E'$  of expressed segments is *supported* by single-end or paired-end reads if there is at least one single-end or paired-end read  $r$  such that  $r[i] = 1, i \in E'$ .

Although single-end reads, paired-end reads, and TSS-PAS information data do not provide exact combinations of expressed segments of isoforms, they can be used to eliminate many isoforms from consideration. Each of these types of data provides some information that can be used to define a condition which will be satisfied by all isoforms inferred by our algorithm (to be described in the next subsection).

- Junction information. A junction  $(s_i, s_j)$  is on an isoform  $f$  if  $f[i] = f[j] = 1$  and  $f[k] = 0, i < k < j$ . If  $s_i$  and  $s_j$  are adjacent, then junction  $(s_i, s_j)$  is an *adjacent junction*. An isoform satisfies *condition I* if all the non-adjacent junctions on this isoform are supported by single-end short reads. In practice, most sufficiently expressed isoforms satisfy this condition. For example, when 40 millions single-end reads with length 30bps are mapped, the probability of an isoform with expression level 6 RPKM satisfying condition I is 99.3% and 92.8%, if this isoform contains 10 and 100 exons, respectively. See Theorem 2 below for the details.

- Start-end segment pair information. For an isoform  $f$ , expressed segment  $s_i$  is the *start* expressed segment of  $f$  if  $f[i] = 1$  and  $f[j] = 0, 1 \leq j < i$ . Expressed segment  $s_i$  is the *end* expressed segment of  $f$  if  $f[i] = 1$  and  $f[j] = 0, i < j \leq |S|$ . The TSS-PAS pair information describes the start and end expressed segments of each isoform and will be referred to as the *start-end segment pair* data. An isoform satisfies *condition II* if the start-end segment pair of this isoform appears in the given set of start-end segment pairs. If the TSS-PAS pair information is missing, then any expressed segment can theoretically be the start or end expressed segment. However, in this case, many short (and thus unrealistic) isoforms could be introduced, which will make isoform inference difficult. Therefore, when the TSS-PAS pair information is missing, we allow an expressed segment  $s_i$  to be the start (or end) expressed segment of any isoform if there is no expressed segment  $s_j$  with  $j < i$  (or  $i < j$ ) such that junction  $(s_j, s_i)$  (or  $(s_i, s_j)$ ), respectively) is adjacent or supported by some read.
- Paired-end read data. A pair of expressed segments  $(s_i, s_j), i < j$  on an isoform  $f$  is an *informative pair* if  $f[i] = f[j] = 1$  and  $P_{M,h,\alpha}(l_i + L_2 - 1, g_{i,j}, l_j + L_2 - 1) < 0.05$ , assuming that the span of a paired-end read follows some probability distribution  $h(x)$ , the expression level of this isoform is  $\alpha$  RPKM and  $M$  paired-end reads have been mapped. Here,  $L_2$  is the read length of a paired-end read,  $g_{i,j} = \sum_{i < k < j} l_k f[k]$ , and  $P_{M,h,\alpha}$  is defined in Theorem □. According to the theorem, if  $(s_i, s_j)$  is informative, then the probability that there are no paired-end reads with start positions in segment  $s_i$  and end positions in segment  $s_j$  is less than 0.05. A triple of expressed segments  $(s_i, s_{i+1}, s_j), i + 1 < j$  is an *informative triple* if  $f[i] = f[i + 1] = f[j] = 1$  and  $P_{M,h,\alpha}(L_2 - 1, g_{i,j}, l_j + L_2 - 1) < 0.05$ . Similarly,  $(s_i, s_{i+1}, s_j), j < i$  is an informative triple if  $P_{M,h,\alpha}(L_2 - 1, g_{j,i+1}, l_j + L_2 - 1) < 0.05$ . An isoform satisfies *condition III* if every informative pair or triple on this isoform is supported by paired-end reads. A larger  $\alpha$  makes this condition more stringent. Because in many cases, two isoforms can only be distinguished by a pair or triple of segments, it is necessary to require that every informative pairs or triple (instead of some of them) are supported by paired-end reads.

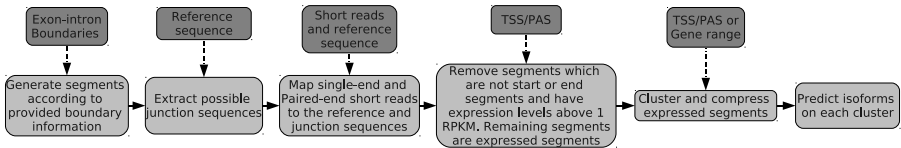
Note that while the junction information is always available given the single-end read data and exon-intron boundary information, the start-end segment pair information and paired-end read data are not necessarily always available. We define an isoform as *valid* if it satisfies conditions I, II and/or III whenever the corresponding types of data are provided. The following theorem gives a lower bound on the probability that type I condition is satisfied by an isoform.

**Theorem 2.** *Under the uniform sampling assumption, the probability that an isoform  $f$  consisting of  $t$  exons with expression level  $x$  RPKM satisfies type I condition is at least  $(1 - e^{-xL_1M/10^9})^{t-1}$ , where  $e$  is the base of natural logarithm,  $M$  the number of single-end reads mapped, and  $L_1$  the length of single-end reads.*

## 2.5 Isoform Inference Algorithm

We now describe our algorithm, IsoInfer, for inferring isoforms. The algorithm uses the following types of data: the reference genome, single-end short reads,

exon-intron boundaries, TSS-PAS pairs, gene boundary information from the reference genome annotation, and paired-end short reads. The first three pieces of information (*i.e.*, the reference genome, exon-intron boundaries and single-end short reads) are required in the algorithm. If TSS-PAS pairs are not provided, gene boundaries would be required. The flow of data processing in IsoInfer is illustrated in Figure 3. The third step of the algorithm requires an external tool (*e.g.*, Bowtie [45]) to map the short reads to the reference genome and junction sequences. In the fifth step, any two segments that are adjacent or supported by a junction read will be clustered together. Note that, such a cluster may contain expressed segments from more than one gene or contain only a subset of expressed segments from a single gene, but these cases do not happen very often. Furthermore, in each cluster, if there is a sequence of consecutive expressed segments such that every internal segment has no non-adjacent junction with any other expressed segment other than its left or right neighbor in the sequence, then we will combine the expressed segments into a single segment. This compression will be important because it reduces the problem size drastically for some isoforms containing a very large number of expressed segments. The details of the clustering and compression step are straightforward and omitted.



**Fig. 3.** The flow of data processing in algorithm IsoInfer

In the following, we give more details of the last step in IsoInfer, *i.e.*, inferring isoforms. Each cluster of expressed segments defines an instance of the isoform inference problem. Denote such an instance as  $I(S, R, T, d)$ , where  $S = \{s_1, s_2, \dots, s_{|S|}\}$  is the set of expressed segments in the cluster,  $R$  the set of short (single-end and paired-end) reads mapped to the segments in the cluster,  $T$  the set of start-end segment pairs, and  $d$  a function such that  $d(i), s_i \in S$ , denotes the number of single-end reads mapped to segment  $s_i$  and  $d(i, j), 1 \leq i < j \leq |S|$ , denotes the number of single-end reads mapped to junction  $(s_i, s_j)$ .

The inference procedure is summarized in Algorithm 1. It first enumerates all the valid isoforms in step 1. However, for a cluster with a large number of expressed segments and isoforms, the number of valid isoforms could be too large to be enumerated efficiently even though conditions I, II and/or III could be used to filter out many invalid isoforms. Therefore, the algorithm enumerates valid isoforms with high expression levels first, where the expression level of an isoform is defined by the least number of single-end reads on any junction of the isoform. The enumeration terminates when a preset number (denoted as  $\gamma$ ) of valid isoforms are enumerated. The parameter  $\gamma$  is used to avoid the rare cases that the number of valid isoforms is too large to be handled by subsequent

steps of IsoInfer. We set  $\gamma = 1000$  by default based on our empirical knowledge of the real data considered in section 4. For example, over 97.5%, 98.5%, and 99% cases, the number of valid isoforms is no more than 1000 in the tests on mouse brain, liver and muscle tissues, respectively, when the exact boundary and TSS-PAS information is extracted from the UCSC knownGene table. The impact of the omitted isoforms is minimized because highly expressed isoforms are enumerated first.

A short read  $r$  is *validated* by a set of isoforms if the set contains an isoform  $f$  such that  $f[i] = 1$  when  $r[i] = 1$ . A start-end segment pair is validated by a set of isoforms if this pair is the start-end segment pair of some  $f$  in the set. A set of isoforms is a *feasible solution* of  $I(S, R, T, d)$  if every read in  $R$  and start-end segment pair in  $T$  are validated by the set. Due to possible noise in sequencing and the incompleteness of the enumeration of valid isoforms in step 1, it may happen that some reads or start-end segment pairs are not supported by the set of isoforms  $F$  enumerated in step 1. Step 2 of the algorithm removes such invalidated reads and start-end segment pairs to make  $F$  feasible.

---

**Algorithm 1.** IsoformInference. Given an instance  $I(S, R, T, d)$ , the algorithm infers a set of isoforms to explain the read data.

---

- 1: Among all segment junctions of an isoform  $f$ , denote  $m(f)$  as the minimum number of single-end reads mapped to any of these junctions. Enumerate all the valid isoforms  $f$  in the descending order of  $m(f)$  until a preset number ( $\gamma$ ) of valid isoforms is obtained. Denote the set of all the enumerated valid isoforms as  $F$ .
  - 2: Remove all the short reads and start-end segment pairs that are not validated by  $F$ .
  - 3: **for**  $5 \leq u \leq \beta$  **do**
  - 4:    $w(f) \leftarrow 0$  for  $f \in F$ .
  - 5:   **for**  $0 \leq m \leq |S| - u$  **do**
  - 6:      $n \leftarrow m + u$ .
  - 7:      $V^{(m,n)} \leftarrow \text{BestCombination}(I^{(m,n)})$ .
  - 8:     For each  $v \in V^{(m,n)}$ , define  $G(v) = \{f | f \in F, f^{(m,n)} = v\}$  and for each  $f \in G(v)$ , let  $w(f) = w(f) + 1/|G(v)|$ .
  - 9:   **end for**
  - 10: Sort  $F$  by  $w$  in increasing order.
  - 11: **for**  $f \in F$  **do**
  - 12:   **if**  $w(f) < 1$  and  $F - \{f\}$  is a feasible solution of  $I$  **then**
  - 13:      $F \leftarrow F - \{f\}$ .
  - 14:   **end if**
  - 15: **end for**
  - 16: **end for**
  - 17:  $w'(f) \leftarrow 1/w(f)$  for  $f \in F$ .
  - 18: Solve the weighted set cover instance  $(U, \mathcal{C}, w')$ , where  $U = R \cup T$ ,  $\mathcal{C} = \{S_f | f \in F\}$ , and  $r \in S_f$  if  $r$  is validated by  $f$  for  $r \in U$  for each  $f \in F$  by the branch-and-bound method implemented in GNU package GLPK. Return the set of the valid isoforms corresponding to the optimal solution of set cover.
-

To find a subset of valid isoforms to explain the data, a simple idea is to try all possible combinations of the valid isoforms in  $F$  and find a minimum combination that can explain all the short reads, as done in procedure *BestCombination* (i.e., Algorithm 2). The procedure *BestCombination* gradually increases the number of valid isoforms considered and enumerates all possible combinations of such a number of isoforms until a preset condition is met.

---

**Algorithm 2.** *BestCombination*. Given an instance  $I(S, R, F, d)$ , find a “best” subset of  $F$  such that the read data can be explained by enumerating all possible subsets of  $F$ .

---

```

1: for  $1 \leq i \leq |S|$  do
2:    $p \leftarrow 0$  and  $F' \leftarrow \emptyset$ .
3:   for each  $F'' \subset F$  where  $|F''| = i$  and  $F''$  is a feasible solution of  $I$  do
4:      $\{z, x\} \leftarrow \text{QP solver}(I(S, F'', d))$ .
5:     if  $p < p\text{-value}(z)$  then
6:        $p \leftarrow p\text{-value}(z)$  and  $F' \leftarrow F''$ .
7:     end if
8:   end for
9:   if  $p \geq 0.05$  then
10:    Return  $F'$ .
11:   end if
12: end for

```

---

However, it is often infeasible to enumerate all possible combinations of the valid isoforms of a given size. When this happens, we decompose an the instance into some sub-instances. In each sub-instance, only a subset of expressed segments are considered. More specifically, for an instance  $I(S, R, F, d)$ , where  $F$  is the set of valid isoforms enumerated, a sub-instance  $I^{(m,n)} = I(S^{(m,n)}, R^{(m,n)}, d^{(m,n)}, F^{(m,n)})$ ,  $0 \leq m < n \leq |S|$ , is defined concerning the subset  $S^{(m,n)} = \{s_{m+1}, \dots, s_n\}$  of expressed segments of  $S$ . It is formally defined as follows. For each  $f \in B(|S|)$ , define  $f^{(m,n)} \in B(n-m)$  and  $f^{(m,n)}[i] = f[i+m]$ ,  $1 \leq i \leq n-m$ . In other words,  $f^{(m,n)}$  denotes the sub-vector of  $f$  spanning the interval  $[m+1, n]$ . Let  $F^{(m,n)} = \{f^{(m,n)} | f \in F\}$ ,  $R^{(m,n)} = \{r^{(m,n)} | r \in R\}$ ,  $d^{(m,n)}(i) = d(i+m)$ ,  $1 \leq i \leq n-m$ , and  $d^{(m,n)}(i, j) = d(i+m, j+m)$ ,  $1 \leq i < j \leq n-m$ . Note that the start-end segment information is not needed in sub-instances.

The parameter  $\beta$  appearing in step 3 controls the maximum size of a sub-instance. Larger sub-instances make the results of procedure *BestCombination* more reliable. However, the execution time of *BestCombination* increases exponentially with the number of valid isoforms which grows with the size of the sub-instance. Therefore, instead of a fixed size, a set of sub-instance sizes from the interval  $[5, \beta]$  are attempted. For a fixed sub-instance size, *BestCombination* is executed on each sub-instance of the size in step 7. According to the results of *BestCombination*, each valid isoform is assigned a weight in Step 8 which roughly indicates the frequency that the isoform appears in the combinations found by *BestCombination*. A subset of valid isoforms with weights less than 1 are removed in steps 11-15 without making  $F$  infeasible.

In steps 17 and 18 of the algorithm, a weighted set cover instance is constructed such that an optimal solution implies a subset of valid isoforms with a minimum total weight such that all the short reads and start-end segments can be explained. The set cover problem can be solved by using the branch-and-bound method implemented in GNU package GLPK, since it involves only small instances.

### 3 Simulation Test Results

We test IsoInfer on mouse genes. The reference genomic sequence and known isoforms of all mouse genes are downloaded from UCSC (mm9, NCBI Build 37) [54]. All exon-intron boundaries of the known isoforms are extracted. This dataset contains 26,989 genes and 49,409 isoforms. 16,392 (60.7%) of the genes have only one isoform and 59 (0.2%) of the genes have more than 10 isoforms. 5830 (21.6%) of the genes have only one exon and 384 (1.4%) of the genes have more than 40 exon-intron boundaries. For the simulation study, only genes with at least two known isoforms are used, which result in 10,595 genes. We further extract all the start-end segments and randomly generate relative expression levels of every isoform. Although it would be natural to assume that expression levels follow a uniform distribution, it is reported in [55,56,57] that the expression levels of isoforms tend to obey a log-normal distribution. Therefore, we consider three types of distributions.

- Base10: For each isoform, a random number  $r$  following the standard normal distribution is generated and then  $10^r$  is assigned as the relative expression level of this isoform.
- Base2: For each isoform, a random number  $r$  following the standard normal distribution is generated and then  $2^r$  is assigned as the relative expression level of this isoform.
- Uniform: For each isoform, a random number  $r$  uniformly generated from  $[0,1]$  is assigned as the relative expression level of this isoform.

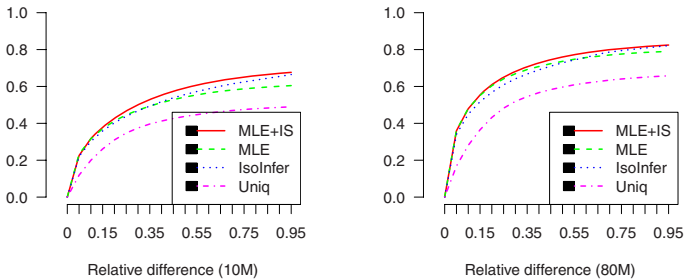
Then 40M single-end and 10M paired-end short reads are randomly generated according to the relative expression levels of the isoforms. In the simulation, we assume that the span of a paired-end read is a random variable obeying the normal distribution  $N(\mu, \sigma^2)$  [58] so we could evaluate the impact of the mean and deviation of the spans of paired-end reads on the performance of IsoInfer. Note that IsoInfer does not depend on this assumption and works for paired-end reads drawn from any distribution.

Finally, IsoInfer is used to recover all the known isoforms using the start-end segments and single-end and paired-end reads. In the simulation, the read lengths of single-end and paired-end reads are 25bps and 20bps, respectively. The parameter  $\alpha$  is set to 1 RPKM,  $\beta = 7$  and  $\gamma = 1000$ . We consider three measures of the performance, *sensitivity*, *effective sensitivity* and *precision*. A known isoform is *recovered* if it is in the output of IsoInfer. Sensitivity is defined as the number of recovered isoforms divided by the number of all known isoforms.

Precision is defined as the number of recovered isoforms divided by the number of isoforms inferred. Since IsoInfer only intends to infer isoforms that are sufficiently expressed, it is useful to consider how many sufficiently expressed isoforms are recovered by the algorithm. Since Theorem 2 shows that an isoform with a sufficiently high expression level is likely to satisfy condition I (*i.e.*, all its exon-intron junctions are supported by the read data) with high probability, we define *effective sensitivity* as the number of recovered isoforms divided by the number of known isoforms whose exon-intron junctions are supported by the read data.

### 3.1 Calculation of Expression Levels

To estimate the effectiveness of our QP formulation, we randomly generate Base10 expression levels and single-end short reads on the known mouse isoforms and check whether it can recover the correct expression levels of the known isoforms. For an isoform  $f$  with expression level  $x_f$  and calculated expression level  $x'_f$ , the relative difference  $\frac{|x'_f - x_f|}{x_f}$  is used to measure the accuracy of calculation. A simple and widely used method of calculating expression levels of isoforms is based on counting reads mapped to its unique exons and exon junctions [41,40]. Clearly, this simple strategy fails if the isoform does not have any unique exons or exon junctions. We compare our method with the simple method (simply denoted as *Uniq* in this paper) and the method based on maximum likelihood estimation (MLE) and importance sampling (IS) [29]. The comparison is depicted in Figure 4.



**Fig. 4.** Comparison of the accuracies of different methods in estimating isoform expression levels. The Y-axis shows the percentage of isoforms whose estimated/calculated expression levels are within a certain relative difference range from the truth. 10 million reads (left) and 80 million reads (right) are sampled in each of the figures.

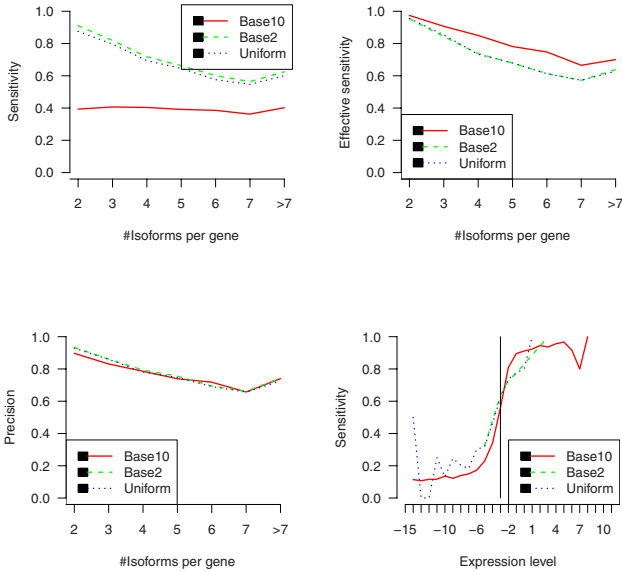
The comparison shows that MLE followed by IS (MLE+IS) is the most accurate and Uniq is the worst. IsoInfer achieves comparable performances with MLE (followed by IS). An advantage of MLE+IS is that it also provides a 95% confidence interval for each expression level estimation. On the other hand, IsoInfer calculates the expression levels much faster than MLE+IS does (3 minutes vs 3 hours for all mouse genes on a standard desktop PC). The efficiency of IsoInfer makes the search for novel isoforms possible.



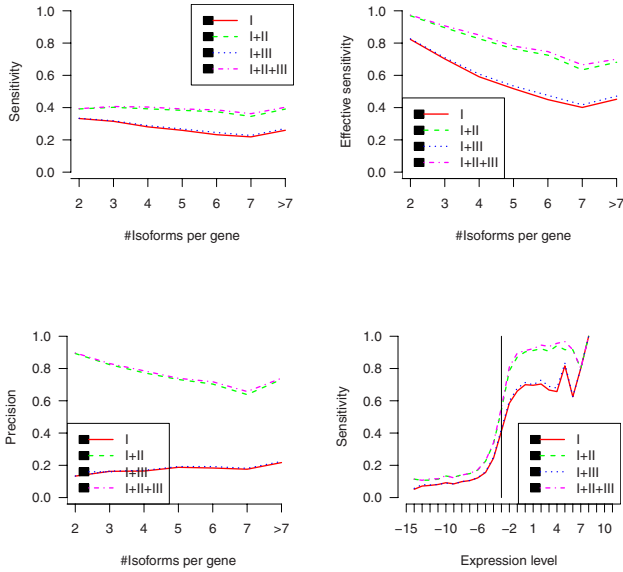
### 3.2 The Influence of the Distribution of Expression Levels

In this section, we analyze the influence of the distribution of expression levels on the performance of IsoInfer in inferring isoforms. The distribution of the span of paired-end reads are fixed as the normal distribution  $N(300, 30^2)$ . The sensitivities and precisions grouped by number of known isoforms per gene are depicted in Figure 5.

The overall sensitivities and precisions of IsoInfer on (Base10, Base2, Uniform) expression levels are (39.7%,75.0%,72.5%) and (79.3%,82.1%,81.3%), respectively. The sensitivities for Base10 expression levels are much lower than those for Base2 and Uniform expression levels, because a large fraction of the isoforms are not significant expressed. The effective sensitivity of three cases are 83.5%, 77.4% and 77.4%, respectively. Figure 5 gives detailed sensitivity, effective sensitivity and precision of IsoInfer on genes with a certain number of isoforms. The high effective sensitivity shown in the figure is also confirmed by the sensitivity results on different expression levels, also given in Figure 5 which shows that isoforms with high expression levels are identified with high sensitivities. For example, for Base10 expression levels, isoforms with expression level above 3 (or 6) RPKM are identified with sensitivity above 56.0% (or 81.0%, respectively).



**Fig. 5.** The sensitivity (top left), effective sensitivity (top right) and precision (bottom left) of IsoInfer on genes with a certain number of isoforms when different distributions of expression levels are generated. The bottom right graph shows the sensitivity of IsoInfer on different expression levels when different distributions of expression level are applied. In the graph, the expression levels are  $\log_2$  transformed. Expression level  $x$  corresponds to  $25 \cdot 2^x$  RPKM. The vertical line corresponds to expression level  $1/8 = 3.125$  RPKM.



**Fig. 6.** The sensitivity (top left), effective sensitivity (top right) and precision (bottom left) of IsoInfer on genes with a certain number of isoforms when different combinations of type I, II and III data are provided. The bottom right graph shows the sensitivity of IsoInfer on different expression levels when different combinations of type I, II and III data are used. Again, the expression levels are  $\log_2$  transformed. Expression level  $x$  corresponds to  $25 \cdot 2^x$  RPKM. The vertical line corresponds to expression level  $1/8 = 3.125$  RPKM.

### 3.3 The Importance of Start-End Expressed Segment Pairs

As mentioned before, single-end short reads are necessary for our algorithm but start-end segment pairs and paired-end reads are optional. To estimate the importance of the last two pieces of information, we compare the results when different types of data are available. Four combinations are possible, denoted as I, I+II, I+III, and I+II+III, where I, II and III correspond to single-end reads (which provide the junction information), start-end segment pairs and paired-end data, respectively. The combination I+III means that the single-end and paired-end read data are available but not the start-end segment pairs. In the simulation, Base10 expression levels are generated and the span distribution of paired-end reads is fixed as  $N(300, 30^2)$ . Figure 6 shows that start-end segment pairs are much more important than paired-end reads for our algorithm. For example, the sensitivities and precisions for combinations I+II and I+III are (38.9%, 78.5%) and (29.5%, 16.5%), respectively.

### 3.4 The Influence of Span Distribution

The span of paired-end reads follows the normal distribution  $N(\mu, \sigma^2)$ . We run IsoInfer on different combinations of  $\mu$  and  $\sigma$ . On each combination, 10 million

pair-end reads are randomly generated. Since start-end segment pairs are much more important than paired-end reads, as shown in the above subsection, the span distribution should not have a significant influence on the inference results when start-end segment pairs are available. This is confirmed by Tables 3 and 4 given in [42]. The precision and sensitivity of IsoInfer vary by at most 1.5% when different span distributions are applied.

The above small effect of paired-end read data on the performance of IsoInfer is because the parameter  $\alpha$  is set to 1. When a large  $\alpha$  is applied, IsoInfer trades sensitivity for precision. For example, when the span distribution of paired-end read is fixed as  $N(300, 30^2)$ , if  $\alpha$  is set to 1, the sensitivity and precision on genes with at least 8 isoforms are 40.2% and 74.0%, respectively. The two measures will change to 35.4% and 78.1%, respectively, when  $\alpha$  is set to 20. The performance of IsoInfer when  $\alpha$  is set to different values is shown in Tables 5 and 6 of [42].

## 4 Recovery of Known Isoforms from Real Reads

The evaluation uses the following four data sets: (1) known mouse isoforms downloaded from UCSC [54], which contains 49,409 transcripts, (2) mouse mRNAs expressed in various tissues downloaded from UCSC containing 228,779 mRNAs, (3) RNA-Seq data from brain, liver and skeletal muscle tissues of mouse [24], which contains 47,781,892, 44,279,807 and 38,210,358 single-end reads for brain, liver and muscle, respectively, and (4) 104,710 exon junctions that were predicted by TopHat from the above RNA-Seq data for mouse brain tissue [19].

As in the simulation tests, on a specific tissue, one can only expect that isoforms with expression levels above a certain threshold can be detected by RNA-Seq experiments, so as to be inferred by IsoInfer. Given a set of mapped reads, an isoform is said to be *theoretically expressed* if each exon except for the first and last one of this isoform has expression level at least 1 RPKM and every exon junction on this isoform is supported by short reads. (Note that this does not really guarantee that the isoform is actually expressed.) The expression levels of the first and last exons are ignored here because of the possible 3' and 5' sampling biases in RNA-Seq [27,24]. The theoretically expressed isoforms among known mouse isoforms and mRNAs are used as benchmarks. Note that the benchmarks change when different tissues are considered, because the expression levels of isoforms change from tissue to tissue.

We have done two group of tests. The first one is to use the TSS-PAS pair and exon-intron boundary information from the known mouse isoforms and/or mRNAs from UCSC and RNA-Seq short reads to infer isoforms. The predicted isoforms are compared with the theoretically expressed isoforms in the corresponding benchmark. An isoform is recovered by IsoInfer if one of isoforms inferred by IsoInfer matches this isoform *precisely* (*i.e.*, the two isoforms contain exactly the same set of exons with exactly the same boundaries). The inference results are shown in Table 1. These results demonstrate that when accurate exon-intron boundary and TSS-PAS pair information is provided, IsoInfer achieves a reasonably good precision, and the precision increases as the size of the benchmark increases. When known mouse isoforms are used, IsoInfer achieves decent

effective sensitivities (*i.e.*, 72.9% for brain, 82.2% for liver and 83.0% for muscle). Because mRNAs were collected from different sources and tissues, a large fraction of them may not really be expressed in a specific tissue. Therefore, effective sensitivity of IsoInfer drops when mRNAs are used as the benchmark.

**Table 1.** The performance of IsoInfer when different exon-intron boundary and TSS-PAS pair information and corresponding benchmarks are used. Here, “Union” means that the exon-intron boundary and TSS-PAS pair information is extracted from both known mouse isoforms and mRNAs and the benchmark is the union of the known mouse isoforms and mRNAs.

Tissue	Known isoforms			mRNAs			Union		
	Brain	Liver	Muscle	Brain	Liver	Muscle	Brain	Liver	Muscle
#Theoretically expressed	18521	12411	11723	87178	72594	69086	101392	82199	78298
Precision	0.493	0.592	0.627	0.572	0.670	0.712	0.591	0.697	0.737
Effective sensitivity	0.729	0.822	0.830	0.328	0.352	0.366	0.335	0.365	0.381

The second test measures the performance of IsoInfer when the exact exon-intron boundary information is unavailable. The test uses exon-intron boundaries predicted by TopHat from the RNA-Seq read data on the mouse brain tissue and the TSS-PAS pair information extracted from the known mouse isoforms and/or mRNAs. The test results are shown in Table 2. Although it is reported in [19] that over 80% of the exon junctions predicted by TopHat are also exon junctions in the UCSC known mouse isoforms, the inference result on the known mouse isoforms is much worse than the result when exact exon-intron boundary information is provided. On the other hand, when mRNA is used as the benchmark, the exon-intron boundaries provided by TopHat lead IsoInfer to a more aggressive prediction (and thus achieving a better effective sensitivity).

In each of the above tests, the last three steps of IsoInfer shown in Figure 3 took less than 80 minutes on an Intel P8600 processor.

**Table 2.** The performance of IsoInfer when the exon-intron boundary information is extracted from the exon junctions predicted by TopHat. These results are all on the mouse brain tissue. The TSS-PAS pair information is extracted from the known mouse isoforms and/or mRNAs, depending on the benchmark. Again, “Union” means that the TSS-PAS pair information is extracted from both known mouse isoforms and the benchmark is the union of the known mouse isoforms and mRNAs.

	Known isoforms	mRNAs	Union
Precision	0.240	0.362	0.378
Effective sensitivity	0.496	0.532	0.508

## Acknowledgment

We thank Pirola Yuri for useful discussions. The research is supported in part by a CSC scholarship, NSF grant IIS-0711129, and NIH grants 2R01LM008991 and AI078885.

## References

1. Boguski, M.S., et al.: Gene discovery in dbEST. *Science* 265(5181), 1993–(1994)
2. Boguski, M.S.: The turning point in genome research. *Trends in Biochemical Sciences* 20(8), 295–296 (1995)
3. The FANTOM Consortium: The transcriptional landscape of the mammalian genome. *Science* 309(5740), 1559–1563 (2005)
4. The ENCODE Project Consortium: Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature* 447(7146), 799–816 (2007)
5. Weinstock, G.M.: ENCODE: more genomic empowerment. *Genome Res.* 17(6), 667–668 (2007)
6. Bertone, P., et al.: Global identification of human transcribed sequences with genome tiling arrays. *Science* 306(5705), 2242–2246 (2004)
7. Kwan, T., et al.: Genome-wide analysis of transcript isoform variation in humans. *Nat. Genetics* (2008)
8. Johnson, J.M., et al.: Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. *Science* 302(5653), 2141–2144 (2003)
9. Kapranov, P., et al.: RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science* 316(5830), 1484–1488 (2007)
10. Brenner, S., et al.: Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat. Biotechnol.* 18(6), 630–634 (2000)
11. Reinartz, J., et al.: Massively parallel signature sequencing (MPSS) as a tool for in-depth quantitative gene expression profiling in all organisms. *Brief Funct. Genomic Proteomic* 1(1), 95–104 (2002)
12. Velculescu, V.E., et al.: Serial analysis of gene expression. *Science* 270(5235), 484–487 (1995)
13. Harbers, M., Carninci, P.: Tag-based approaches for transcriptome research and genome annotation. *Nat. Meth.* 2(7), 495–502 (2005)
14. Shiraki, T., et al.: Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proceedings of the National Academy of Sciences of the United States of America* 100(26), 15776–15781 (2003)
15. Kodzius, R., et al.: CAGE: cap analysis of gene expression. *Nat. Meth.* 3(3), 211–222 (2005)
16. Kim, J.B., et al.: Polony multiplex analysis of gene expression (PMAGE) in mouse hypertrophic cardiomyopathy. *Science* 316(5830), 1481–1484 (2007)
17. Ng, P., et al.: Gene identification signature (GIS) analysis for transcriptome characterization and genome annotation. *Nat. Methods* 2, 105–111 (2005)
18. Nagalakshmi, U., et al.: The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320(5881), 1344–1349 (2008)
19. Trapnell, C., et al.: TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25(9), 1105–1111 (2009)
20. Graveley, B.R.: Molecular biology: power sequencing. *Nature* 453(7199), 1197–1198 (2008)
21. Yassour, M., et al.: Ab initio construction of a eukaryotic transcriptome by massively parallel mRNA sequencing. *Proceedings of the National Academy of Sciences* 106(9), 3264–3269 (2009)
22. Wilhelm, B.T., et al.: Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature* 453(7199), 1239–1243 (2008)

23. Cloonan, N., et al.: Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nat. Methods* (2008)
24. Mortazavi, A., et al.: Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods* 5(7), 621–628 (2008)
25. Marioni, J., et al.: RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.* 18(9), 1509–1517 (2008)
26. Sultan, M., et al.: A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* 321(5891), 956–960 (2008)
27. Wang, Z., et al.: RNA-Seq: a revolutionary tool for transcriptomics. *Genetics Nature reviews* (2008)
28. Lacroix, V., et al.: Exact transcriptome reconstruction from short sequence reads. In: Crandall, K.A., Lagergren, J. (eds.) *WABI 2008*. LNCS (LNBI), vol. 5251, pp. 50–63. Springer, Heidelberg (2008)
29. Jiang, H., Wong, W.H.: Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics* 25(8), 1026–1032 (2009)
30. Pagani, F., Baralle, F.E.: Genomic variants in exons and introns: identifying the splicing spoilers. *Nat. Rev. Genet.* 5(5), 389–396 (2004)
31. Srebrow, A., Kornblihtt, A.R.: The connection between splicing and cancer. *J. Cell Sci.* 119(13), 2635–2641 (2006)
32. Williams, W.V.: Editorial hot topic: Transcriptome analysis in drug development (executive editor: williams, W.v.). *Current Molecular Medicine* 5(2), 1–2 (2005)
33. Heber, S., et al.: Splicing graphs and EST assembly problem. *Bioinformatics* 18(suppl.1), S181–S188 (2002)
34. Sammeth, M., Valiente, G., Guigó, R.: Bubbles: Alternative splicing events of arbitrary dimension in splicing graphs. In: Vingron, M., Wong, L. (eds.) *RECOMB 2008*. LNCS (LNBI), vol. 4955, pp. 372–395. Springer, Heidelberg (2008)
35. Xing, Y., et al.: The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures. *Genome Res.* 14(3), 426–441 (2004)
36. Bonizzoni, P., et al.: Detecting alternative gene structures from spliced ESTs: a computational approach. *Journal of Computational Biology* 16(1), 43–66 (2009)
37. Djebali, S., et al.: Efficient targeted transcript discovery via array-based normalization of RACE libraries. *Nat. Meth.* 5(7), 629–635 (2008)
38. Salehi-Ashtiani, K., Yang, X., Derti, A., Tian, W., Hao, T., Lin, C., Makowski, K., Shen, L., Murray, R.R., Szeto, D., Tusneem, N., Smith, D.R., Cusick, M.E., Hill, D.E., Roth, F.P., Vidal, M.: Isoform discovery by targeted cloning, ‘deep-well’ pooling and parallel sequencing. *Nat. Meth.* 5(7), 597–600 (2008)
39. Fullwood, M.J., et al.: Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses. *Genome Res.* 19(4), 521–532 (2009)
40. Pan, Q., et al.: Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat. Genet.* 40(12), 1413–1415 (2008)
41. Wang, E.T., et al.: Alternative isoform regulation in human tissue transcriptomes. *Nature* 456(7221), 470–476 (2008)
42. Feng, J., et al.: Inference of isoforms from short sequence reads. Manuscript (January 2010), <http://www.cs.ucr.edu/~jianxing/IsoInfer-recomb10-full.pdf>
43. Breitbart, R.E., et al.: Alternative splicing: a ubiquitous mechanism for the generation of multiple protein isoforms from single genes. *Annual Review of Biochemistry* 56(1), 467–495 (1987)
44. Sammeth, M., et al.: A general definition and nomenclature for alternative splicing events. *PLoS Comput. Biol.* 4(8), e1000147 (2008)

45. Langmead, B., et al.: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10(3), R25 (2009)
46. Li, H., et al.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* 18(11), 1851–1858 (2008)
47. Li, R., et al.: SOAP: short oligonucleotide alignment program. *Bioinformatics* 24(5), 713–714 (2008)
48. Cloonan, N., et al.: RNA-MATE: a recursive mapping strategy for high-throughput RNA-sequencing data. *Bioinformatics*, btp459 (2009)
49. Alkan, C., Kidd, J.M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J.O., Baker, C., Malig, M., Mutlu, O., Sahinalp, S.C., Gibbs, R.A., Eichler, E.E.: Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.* 41(10), 1061–1067 (2009)
50. Hashimoto, T., et al.: Probabilistic resolution of multi-mapping reads in massively parallel sequencing data using MuMRescueLite. *Bioinformatics*, btp438 (2009)
51. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2007)
52. Goldfarb, D., Idnani, A.: A numerically stable dual method for solving strictly convex quadratic programs. *Math. Program* 27, 1–33 (1983)
53. Korbel, J., et al.: PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology* 10(2), R23 (2009)
54. Karolchik, D., et al.: The UCSC genome browser database: 2008 update. *Nucl. Acids Res.* 36(Database issue), D773–D779 (2008)
55. Alter, M.D., et al.: Variation in the large-scale organization of gene expression levels in the hippocampus relates to stable epigenetic variability in behavior. *PLoS ONE* 3(10), e3344 (2008)
56. Konishi, T.: Three-parameter lognormal distribution ubiquitously found in cdna microarray data and its application to parametric data treatment. *BMC Bioinformatics* 5(1), 5 (2004)
57. Wijaya, E., et al.: Modeling the marginal distribution of gene expression with mixture models. In: *FGCN 2008: Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking*, pp. 84–89. IEEE Computer Society, Washington (2008)
58. Richter, D.C., et al.: MetaSima sequencing simulator for genomics and metagenomics. *PLoS ONE* 3(10), e3373 (2008)

# The Clark Phase-able Sample Size Problem: Long-Range Phasing and Loss of Heterozygosity in GWAS

Bjarni V. Halldórsson<sup>1,3,4,\*,\*\*</sup>, Derek Aguiar<sup>1,2,\*\*,\*\*\*</sup>,  
Ryan Tarpine<sup>1,2,\*\*,\*\*\*</sup>, and Sorin Istrail<sup>1,2,\*,\*\*</sup>

<sup>1</sup> Center for Computational Molecular Biology, Brown University

`bjarnivh@ru.is`

<sup>2</sup> Department of Computer Science, Brown University

`sorin@cs.brown.edu`

<sup>3</sup> School of Science and Engineering, Reykjavik University

<sup>4</sup> deCODE genetics

**Abstract.** A phase transition is taking place today. The amount of data generated by genome resequencing technologies is so large that in some cases it is now less expensive to repeat the experiment than to store the information generated by the experiment. In the next few years it is quite possible that millions of Americans will have been genotyped. The question then arises of how to make the best use of this information and jointly estimate the haplotypes of all these individuals. The premise of the paper is that long shared genomic regions (or tracts) are unlikely unless the haplotypes are identical by descent (IBD), in contrast to short shared tracts which may be identical by state (IBS). Here we estimate for populations, using the US as a model, what sample size of genotyped individuals would be necessary to have sufficiently long shared haplotype regions (tracts) that are identical by descent (IBD), at a statistically significant level. These tracts can then be used as input for a Clark-like phasing method to obtain a complete phasing solution of the sample. We estimate in this paper that for a population like the US and about 1% of the people genotyped (approximately 2 million), tracts of about 200 SNPs long are shared between pairs of individuals IBD with high probability which assures the Clark method phasing success. We show on simulated data that the algorithm will get an almost perfect solution if the number of individuals being SNP arrayed is large enough and the correctness of the algorithm grows with the number of individuals being genotyped.

We also study a related problem that connects copy number variation with phasing algorithm success. A loss of heterozygosity (LOH) event is when, by the laws of Mendelian inheritance, an individual should be heterozygote but, due to a deletion polymorphism, is not. Such polymorphisms are difficult to detect using existing algorithms, but play an

---

\* Corresponding authors.

\*\* Contributed equally to this work.

\*\*\* Member of the International Multiple Sclerosis Genetics Consortium GWAS Analysis team.



important role in the genetics of disease and will confuse haplotype phasing algorithms if not accounted for. We will present an algorithm for detecting LOH regions across the genomes of thousands of individuals. The design of the long-range phasing algorithm and the Loss of Heterozygosity inference algorithms was inspired by analyzing of the Multiple Sclerosis (MS) GWAS dataset of the International Multiple Sclerosis Consortium and we present in this paper similar results with those obtained from the MS data.

## 1 Introduction

Genome-wide association studies (GWAS) proceed by identifying a number of individuals carrying a disease or trait and comparing these individuals to those that do not or are not known to carry the disease/trait. Both sets of individuals are then genotyped for a large number of Single Nucleotide Polymorphism (SNP) genetic variants which are then tested for association to the disease/trait. GWAS have been able to successfully identify a very large number of polymorphism associated to disease ([19, 4, 11] etc.) and the amount of SNP data from these studies is growing rapidly. Studies using tens of thousands of individuals are becoming commonplace and are increasingly the norm in the association of genetic variants to disease [5, 19, 13]. These studies generally proceed by pooling together large amounts of genome-wide data from multiple studies, for a combined total of tens of thousands of individuals in a single meta-analysis study. It can be expected that if the number of individuals being genotyped continues to grow, hundreds of thousands, if not millions, of individuals will soon be studied for association to a single disease or trait.

SNPs are the most abundant form of variation between two individuals. However, other forms of variation exist such as copy number variation – large scale chromosomal deletions, insertions, and duplications (CNV). These variations, which have shown to be increasingly important and an influential factor in many diseases [17], are not probed using SNP arrays. A further limitation of SNP arrays is that they are designed to probe only previously discovered, common variants. Rare variants, belonging perhaps only to a small set of carriers of a particular disease and hence potentially more deleterious, will not be detected using SNP arrays.

To reach their full potential, the future direction of genetic association studies are mainly twofold: the testing of more individuals using genome-wide association arrays and the resequencing of a small number of individuals with the goal of detecting more types of genetic variations, both rare SNPs and structural variation [16]. Testing multiple individuals for the same variants using standard genome-wide association arrays is becoming increasingly common and can be done at a cost of approximately \$100 per individual. In the next couple of years it is plausible that several million individuals in the US population will have had their genome SNP arrayed. In contrast, whole genome resequencing is currently in its infancy. A few people have had their genome resequenced and the cost of sequencing a single individual is still estimated in the hundreds of thousands of

dollars. However, whole genome sequencing is preferable for association studies as it allows for the detection of all genomic variation and not only SNP variation.

Due to the fact whole genome SNP arrays are becoming increasingly abundant and whole genome resequencing is still quite expensive, the question has been raised whether it would suffice to whole genome sequence a small number of individuals and then impute [7] other genotypes using SNP arrays and the shared inheritance of these two sets of individuals. It has been shown – in the Icelandic population with a rich pedigree structure known – that this could be done most efficiently using the haplotypes shared by descent between the individuals that are SNP arrayed and those that have been resequenced [10]. Haplotype sharing by descent occurs most frequently between closely related individuals, but also occurs with low probability between individuals that are more distantly related. In small closely related populations, as in the Icelandic population, only a moderately sized sample size is therefore needed in order for each individual to have, with high probability, an individual that is closely related to it. In larger populations, such as the US population, a larger sample size will be needed for there to be a significant probability of an individual sharing a haplotype by descent within the population. We say that an individual is “Clark phaseable” with respect to a population sample if the sample contains an individual that shares a haplotype with this individual by descent. In this paper we explore what the required sample size is so that most individuals within the sample are Clark phaseable, when the sample is drawn from a large heterogeneous population, such as the US population.

*Problem 1.* Current technologies, suitable for large-scale polymorphism screening, only yield the genotype information at each SNP site. The actual haplotypes in the typed region can only be obtained at a considerably high experimental cost or computationally by haplotype phasing. Due to the importance of haplotype information for inferring population history and for disease association, the development of algorithms for detecting haplotypes from genotype data has been an active research area for several years [3, 15, 18, 14, 10, 6]. However, algorithms for determining haplotype phase are still in their infancy after about 15 years of development (e.g. [3, 18, 9]). Of particular worry is the fact that the learning rate of the algorithm, i.e. the rate that the algorithms are able to infer more correct haplotypes, grows quite slowly with the number of individuals being SNP arrayed.

*Solution 1.* In this paper we present an algorithm for the phasing of a large number of individuals. We show that the algorithm will get an almost perfect solution if the number of individuals being SNP arrayed is large enough and the correctness of the algorithm grows with the number of individuals being genotyped. We will consider the problem of haplotype phasing from long shared genomic regions (that we call tracts). Long shared tracts are unlikely unless the haplotypes are identical by descent (IBD), in contrast to short shared tracts which may be identical by state (IBS). We show how we can use these long shared tracts for haplotype phasing.

*Problem 2.* We further consider the problem of detecting copy number variations from whole genome SNP arrays. A loss of heterozygosity (LOH) event is when, by the laws of Mendelian inheritance, an individual should be heterozygote but due to a deletion polymorphism, is not. Such polymorphisms are difficult to detect using existing algorithms, but play an important role in the genetics of disease [17] and will confuse haplotype phasing algorithms if not accounted for.

*Solution 2.* We provide an exact exponential algorithm and a greedy heuristic for detecting LOH regions.

For this paper, we run empirical tests and benchmark the algorithms on a simulated GWAS datasets [8] resembling the structure of the International Multiple Sclerosis Genetics Consortium [4] data. To determine LOH events we assume the data is given in trios, i.e. the genotypes of a child and both its parents are known.

## 2 Long Range Phasing and Haplotype Tracts

The haplotype phasing problem asks to computationally determine the set of haplotypes given a set of individual's genotypes. We define a *haplotype tract* (or *tract* for short) denoted  $[i, j]$  as a sequence of SNPs that is shared between at least two individuals starting at the same position  $i$  in all individuals and ending at the same position  $j$  in all individuals. We show that if we have a long enough tract then the probability that the sharing is IBD is close to 1. Multiple sharing of long tracts further increases the probability that the sharing corresponds to the true phasing.

### 2.1 Probability of Observing a Long Tract

We show that as the length of the tract increases the probability that the tract is shared IBD increases. Let  $t$  be some shared tract between two individual's haplotypes and  $l$  be the length of that shared tract. We can then approximate the probability this shared tract is identical by state (IBS)  $p_{IBS}(l)$ . Let  $f_{M,i}$  be the major allele frequency of the SNP in position  $i$  in the shared tract  $t$ . Assuming the Infinite Sites model and each locus is independent,

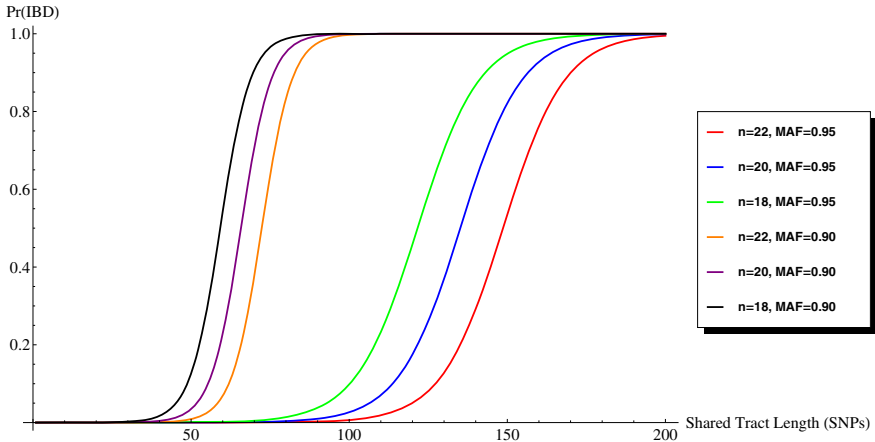
$$p_{IBS}(l) = \prod_{i=1}^l ((f_{M,i})(f_{M,i}) + (1 - f_{M,i})(1 - f_{M,i}))$$

We can approximate  $p_{IBS}(l)$  by noticing  $f_{M,i} * f_{M,i}$  dominates  $(1 - f_{M,i})(1 - f_{M,i})$  as  $f_{M,i} \rightarrow 1$ ,  $p_{IBS}(l) \approx \prod_{i=1}^l (f_{M,i})^2$ . Let  $f_{avg}$  be  $\frac{1}{l} f_{M,i} \forall i \in t$ . Then  $p_{IBS}(l) \approx (f_{avg})^{2l}$ . Given  $f_{M,i}$  is some high frequency, say 95%, then a sharing of 100 consecutive alleles is very unlikely,  $p_{IBS}(100) \approx 0.95^{200} = 10^{-5}$ . For very large datasets we will need to select the length of the tract being considered to be large enough so that the probability that the sharing is identical by state is small.

The probability two individuals separated by  $2(k + 1)$  meiosis ( $k$ th-degree cousins) share a locus IBD is  $2^{-2k}$  [10]. As  $k$  increases, the probability  $k$ th-degree cousins share a particular locus IBD decreases exponentially. However, if two individuals share a locus IBD then they are expected to share about  $\frac{200}{2k+2}$  cM [10]. Relating  $P(IBD)$  to length of tract  $l$ ,

$$P(IBD|sharing\ of\ length\ l) = \frac{2^{-2n}}{2^{-2n} + ((f_{M,i})^{2l} + (1 - f_{M,i})^{2l})}$$

which is shown in Fig. 1



**Fig. 1.** Probability of IBD as a function of shared tract length (measured in SNPs) and plotted for several  $n$  and major allele frequencies (MAF).  $n$  is the number of meiosis between the two individuals. The smaller the MAF or  $n$  the faster  $P(IBM)$  converges to 1.

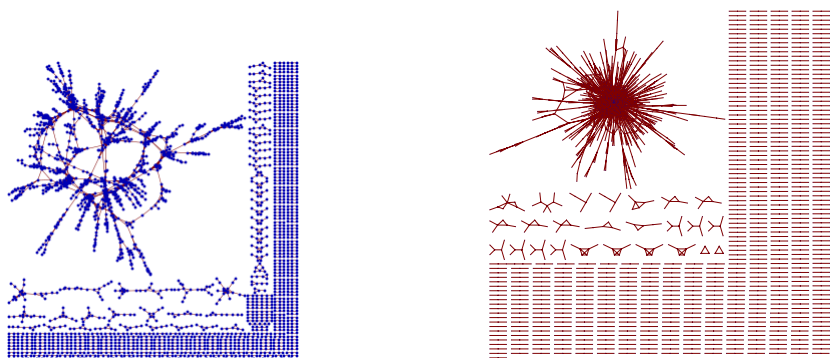
### 2.2 The Clark Phase-Able Sample Size Problem

Given the large tract sharing, we can construct the *Clark consistency graph* having individuals as vertices and an edge between two individuals if they share a tract [15]. Figure 2 shows the Clark consistency graph for different *minimum significant tract lengths* (or window sizes) in the MS dataset. At what minimum significant tract lengths will the graph become dense enough so that phasing can be done properly? What percentage of the population needs to be genotyped so that the Clark consistency graph becomes essentially a single connected component? We call this “The Clark sample estimate: the size for which the Clark consistency graph is connected, C.”

We computed the average number of edges in the haplotype consistency graph as a function of window size to get a sense when the Clark consistency graph of the MS data becomes connected. Based on Fig. 3 and  $P(IBM)$  we can propose an algorithmic problem formulation from the Clark consistency graph. Preferably we would like to solve either one of the below problems.

*Problem 3.* Remove the minimum number of the edges from the Clark consistency graph so that the resulting graph gives a consistent phasing of the haplotypes.

*Problem 4.* Maximize the joint probability of all the haplotypes given the observed haplotype sharing.



**Fig. 2.** Left: The Clark consistency graph for region [1400,1600). A large fraction of individuals share consistent haplotypes of length 200 suggesting many are IBD. Right: The Clark consistency graph for a smaller window size of 180 base pairs. We observe a more dense connected component in part due to the smaller windows size but also because of the specific genomic region.

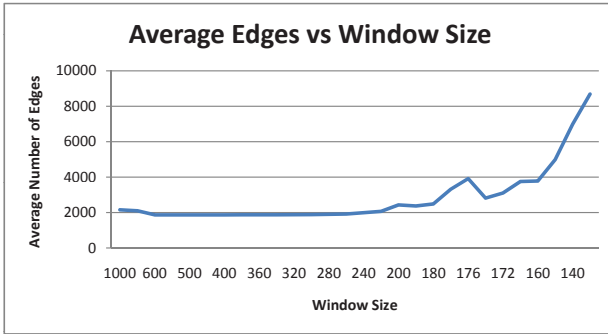
We believe that both of these problem formulations are NP-hard and instead propose to solve these problems using a heuristic. Our benchmarking on simulated data shows that this heuristic works quite well.

### 2.3 Phasing the Individuals That Are Part of the Largest Component

We now proceed with an iterative algorithm working on the connected components in the Clark haplotype consistency graph. First we construct the graph according to some length of haplotype consistency (Fig. 3 and  $P(IBM)$  help define this length). We iterate through each site of each individual to find the tracts. After we find a site with some long shared region, we look at its neighbors in the connected component and apply a voting scheme to decide what the value is for each heterozygous allele. After each individual has been processed we iterate with having resolved sites in the original matrix.

**Observation 1.** *If the Clark consistency graph is fully connected all edges are due to IBD sharing and all individuals can be perfectly phased up to the point where all individuals are heterozygote at a particular site.*

Therefore, phasing individuals in a connected component of the graph should be easy, but in practice there will be some inconsistencies for a number of reasons.



**Fig. 3.** The average number of edges per window size stays relatively constant until a window size of about 180. The graph becomes more connected at this point likely because the window size is small enough to not be largely affected by recombination (but still large enough for the shared tracts to not likely be IBS).

If a node in the Clark consistency graph has a high degree then the phasing of that node will be ambiguous if its neighbors are not consistent. At some times this may be due to genotyping error and at times this may be due to identical by state sharing to either one or both of an individuals haplotypes. The identical by state sharing may because the haplotype has undergone recombination, possibly a part of the haplotype is shared identical by descent and a part is identical by state.

Our alphabet for genotype data is  $\Sigma = \{0, 1, 2, 3\}$ . 0s and 1s represent the homozygote for the two alleles of a SNP. A 2 represents a heterozygous site and a 3 represents missing data. Given a set of  $n$ -long genotype strings  $G = \{g_1, g_2, \dots, g_{|G|}\}$  where  $g_i \in \Sigma^n$ , we represent this in a matrix  $M$  with  $m = 2|G|$  rows and  $n$  columns:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,n} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m,1} & M_{m,2} & \cdots & M_{m,n} \end{bmatrix}$$

Each genotype  $g_i$  is represented by the two rows  $2i-1$  and  $2i$ . Initially,  $M_{2i-1,j} = M_{2i,j} = g_i[j]$ .

We define allele consistency to be:

$$c(a, b) = \begin{cases} 1 & \text{if } a = b \text{ or } a \in \{2, 3\} \text{ or } b \in \{2, 3\} \\ 0 & \text{otherwise} \end{cases}$$

Rows  $r$  and  $s$  of  $M$  are consistent along a tract  $[i, j]$  (i.e. have a shared tract) is written

$$C_{[i, j]}(r, s) = \prod_{k \in [i, j]} c(M_{r,k}, M_{s,k})$$

The length of a tract is written  $|[i, j]| = j - i + 1$ .

A shared tract  $[i, j]$  between rows  $r$  and  $s$  is *maximal shared tract* if it cannot be extended to the left or right; i.e.,  $i = 1$  or  $c(M_{r,i-1}, M_{s,i-1}) = 0$  and  $j = n$  or  $c(M_{r,j+1}, M_{s,j+1}) = 0$ . The maximal shared tract between rows  $r$  and  $s$  at position  $i$  is written  $S_i^{r,s}$ . It is unique. Note that if  $S_i^{r,s} = [j, k]$  then  $\forall l \in [j, k] S_l^{r,s} = S_i^{r,s}$ .

### 2.4 Tract Finding and Phasing Algorithm

Given that there are some loci for which individuals share IBD and that these sharings are expected to be large, we developed an algorithm to detect and use these sharings to resolve the phase at heterozygous sites. Each site is resolved by determining if there are any other individuals that likely share a haplotype by descent. SNPs that do not have their phase determined during any given iteration will be processed in succeeding iterations. If there are enough long IBD loci, this algorithm should unambiguously determine the phase of each individual.

If we know that the data contains trios, a child and both of its parents, we start by phasing the trios using Mendelian laws of inheritance. This replaces many of the heterozygote sites (whenever at least one member of a family is homozygous) and even a few of the sites having missing data (i.e., when the parents are both homozygous and the child’s genotype is missing).

To phase using long shared tracts, we start by fixing a minimum significant tract length  $L$ . We run several iterations, each of which generate a modified matrix  $M'$  from  $M$ , which is then used as the basis for the next iteration.

First, we set  $M' := M$ .

For each row  $r$  we examine position  $i$ . If  $M_{r,i} \in \{0, 1\}$  then we move to the next  $i$ . Otherwise  $M_{r,i} \in \{2, 3\}$ , and we count “votes” for whether the actual allele is a 0 or 1.

$$V_0^r = |\{s \mid s \neq r \text{ and } |S_i^{r,s}| \geq L \text{ and } M_{s,i} = 0\}|$$

$V_1^r$  is defined analogously (the difference being the condition  $M_{s,i} = 1$ ). If  $V_0^r > V_1^r$  then we set  $M'_{r,i} := 0$ . Similarly for  $V_1^r > V_0^r$ . If  $V_0^r = V_1^r$  then we do nothing.

A more complex case is when  $M_{r,i} = 2$ . We make sure the complementary haplotypes are given different alleles by setting the values of both haplotypes simultaneously. This does not cause a dependency on which haplotype is visited first because we have extra information we can take advantage of. We count votes for the complementary haplotype and treat them oppositely. That is, votes for the complementary haplotype having a 1 can be treated as votes for the current haplotype having a 0 (and vice versa). So letting  $r'$  be the row index for the complementary haplotype, we actually compare  $V_0^r + V_1^{r'}$  and  $V_1^r + V_0^{r'}$ . This is helpful when SNPs near position  $i$  (which therefore will fall within shared tracts involving  $i$ ) have already been phased (by trio pre-phasing or previous iterations). It also helps in making the best decision when both haplotypes receive a majority of votes for the same allele, e.g., both have a majority of votes for 0. In this case, taking into account votes for the two haplotypes simultaneously will result in whichever has *more* votes getting assigned the actual value 0. If they

each receive the exact same number of votes, then no allele will be assigned. This also avoids the above-mentioned dependency on the order in which the haplotypes are visited – the outcome is the same since votes for both are taken into account.

In this manner,  $M'$  is calculated at each position. If  $M' = M$  (i.e. no changes were made) then the algorithm terminates. Otherwise,  $M := M'$  ( $M$  is replaced by  $M'$ ) and another iteration is run.

## 2.5 Phasing the Individuals That Are Not a Part of the Largest Component

Individuals that are part of small connected components will have a number of ambiguous sites once they have been phased using the edges in their connected component. For these individuals, we compute a minimum number of recombinations and mutations from their haplotypes to others that have better phasing (belong to larger components). We then assign these haplotypes phase based on minimizing the number of mutations plus recombinations in a similar manner as the approach of Minichiello Durbin [12].

Alternatively this could be done in a sampling framework, where we sample the haplotype with a probability that is a function of the number of mutations and recombinations.

## 2.6 Experimental Results on Simulated Data

We compared the correctness and learning rate of our algorithm against BEAGLE [2] using a simulated dataset. Using the Hudson Simulator [8], we generated 3000 haplotypes each consisting of 3434 SNPs from chromosomes of length  $10^5$ . We estimated a population size of  $10^6$  with a neutral mutation rate of  $10^{-9}$ . To generate genotypes, we randomly sampled from the distribution of simulated haplotypes with replacement such that each haplotype was sampled on average 2, 3, and 4 times. We applied our algorithm and BEAGLE to the simulated data after combining haplotypes to create parent-offspring trio data (inspired by our analysis of the MS dataset). Both algorithms effectively phase the simulated dataset largely due to the initial trio phasing (Table 1). Our algorithm learns the true phasing at an increasing rate as the expectation of haplotypes sampled increases. The most clear example of this trend is in the Brown Long Range Phasing miscall rate. By weighing edges proportional to probability of sharing IBD rather than a fixed set of votes per edge, we should achieve more accurate phasings (subject of future work).

## 3 Loss of Heterozygosity Regions

We call the loss of the normal allele a Loss of Heterozygosity (LOH) which may be a genetic determinant in the development of disease [11, 17]. In some situations, individuals that are heterozygous at a particular locus can possess



**Table 1.** We created three populations using a base pool of 3000 simulated haplotypes using the Hudson simulator. Populations 1, 2, and 3 were created by sampling each haplotype according to a geometric distribution with expectation 2, 3, and 4 respectively. Haplotypes were then randomly paired to create genotypes. The miscall rate is the ratio of 2's miscalled to total 2's (after trio phasing). Error-free phasings denote the number of haplotype phasings with zero miscalled 2's.

	Population 1	Population 2	Population 3
BEAGLE miscall rate	0.0685%	0.0160%	0.00951%
Brown Long Range Phasing miscall rate	0.0501%	0.0148%	0.00503%
BEAGLE Error-free phasings	4467	6819	8898
Brown Long Range Phasing Error-free phasings	4459	6840	8923
Total haplotypes	4524	6870	8940

one normal allele and one deleterious allele. The detection of CNVs, such as deletions, is an important aspect of GWAS to find LOH events, and yet, it is commonly overlooked due to technological and computational limitations.

LOH can be inferred using data from SNP arrays. The SNP calling algorithm for SNP arrays cannot distinguish between an individual who is homozygous for some allele  $a$  and an individual who has a deletion haplotype and the allele  $a$  (Fig. 4 Left). LOH events can then be inferred by finding such genotypic events throughout the dataset. We will present two algorithms for computing putative LOH regions across GWAS datasets.

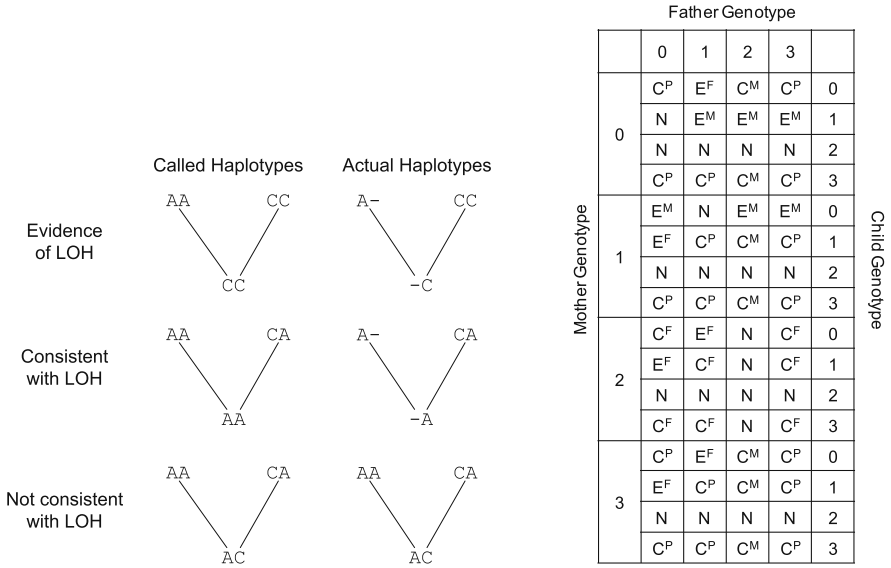
### 3.1 Definitions

A *trio* consists of three individual's genotypes and is defined by the inheritance pattern of parents to child. As before, let  $M$  denote the matrix of genotypes but we now assume  $M$  consists of trios. Let  $M_i$  denote the  $i^{\text{th}}$  trio of  $M$  (individuals  $i$ ,  $i + 1$ , and  $i + 2$ ). At any site  $j$  the trio  $M_i$  may have  $4^3$  possible genotype combinations for which the trio can either be *consistent with LOH* (CLOH), *not consistent with LOH* (NCLOH), or show *evidence of LOH* (ELOH) (Fig. 4 Left). A trio at site  $i$  shows ELOH if the inheritance pattern can only be explained with the use of a deletion haplotype (or a genotyping error). A trio at site  $i$  is NCLOH if the inheritance pattern cannot be explained with the use of a deletion haplotype, and CLOH if it *may* be explained with the use of a deletion haplotype.

### 3.2 The LOH Inference Problem

We are given a set of  $n$  SNPs and a set of  $m$  trios genotyped at those SNPs. For each SNP/trio pair the SNP can have one of three labels:

- X – The marker is inconsistent with having a loss of heterozygosity (Fig. 4 Left: Not Consistent with LOH).



**Fig. 4.** Left: Three examples of inheritance patterns in GWAS data in the context of LOH. The Evidence of LOH (ELOH) pattern shows strong correlation between LOH and a SNP site because the only possible explanation involves introducing a deletion haplotype. An inheritance pattern is called consistent with LOH (CLOH) if it does not contradict the presence of a deletion haplotype and can be explained with normal inheritance patterns. An inheritance pattern not consistent with LOH (NCLOH) occurs when a deletion haplotype cannot be introduced to explain the trio inheritance pattern. Right: The correlation between inheritance pattern and ELOH, CLOH, and NCLOH. We define  $E$  to be ELOH,  $C$  to be CLOH, and  $N$  to be NCLOH. The superscript defines for which parent the putative deletion haplotype is associated. We define the superscript  $F$  to be consistent with a deletion haplotype inherited from the father,  $M$  for mother, and  $P$  for both parents.

- 0 – The marker is consistent with having a loss of heterozygosity (Fig. 4, Left: Consistent with LOH).
- 1 – The SNP shows evidence of loss of heterozygosity, (Fig. 4, Left: Evidence of LOH).

For any trio  $M_i$ , a contiguous sequence of at least one 1 and an unbounded number of 0 sites is called a *putative deletion*. We call two putative deletions,  $p_i$  and  $p_j$ , overlapping if they share at least 1 common index. Let  $h_i$  and  $h_j$  be two ELOH and let  $p_i$  and  $p_j$  contain  $h_i$  and  $h_j$  respectively. Each putative deletion is associated with an interval which is defined by their start and end indices:  $[s_i, e_i]$  and  $[s_j, e_j]$  respectively.  $h_i$  and  $h_j$  are called compatible (or overlapping) if  $h_i$  and  $h_j$  are members of the same putative deletion (i.e.  $h_i \in [s_i, e_i]$  and  $h_j \in [s_i, e_i]$ ) or  $h_i$  is contained in the interval defining  $p_j$  and  $h_j$  is contained in the interval defining  $p_i$ . All CLOH and ELOH sites within a putative deletion

must share the same parent (Fig. 4 Right). The task is to call all 1's  $\in M$  either a deletion or a genotyping error according to some objective function which weighs the relative costs of calling genotyping errors or deletions.

### 3.3 LOH Inference Algorithms

We present an exponential algorithm and a greedy heuristic for computing putative deletions. Both algorithms begin by parsing  $M$  and removing SNPs in which the Mendelian error rate is above 5% to remove artifacts from genotyping. We then calculate the LOH site vector for each trio in the dataset which corresponds to using the table defined in Fig. 4 (Right) to translate each SNP site. This new matrix is denoted  $N^{(\frac{Ml}{3} \times l)}$ . To identify the genotyping errors and putative deletions, we define two operations on  $N$ : error correction call and deletion haplotype call. An error correction call will categorize an ELOH site as a genotyping error effectively removing it from any particular deletion haplotype. An deletion haplotype call will identify a putative deletion as an inherited deletion haplotype. We infer inherited deletion haplotypes using the objective function

$$\min_N (k_1 * (\text{genotype error corrections calls}) + k_2 * (\text{deletion haplotypes calls}))$$

where  $k_1$  and  $k_2$  are weighing factors.  $k_1$  and  $k_2$  can be simple constant factors or a more complex distribution. For example, setting  $k_1$  to 2 and  $k_2$  to 7, we will prefer calling a putative deletion with at least 4 pairwise compatible ELOH sites an inherited deletion. For a more complex objective function, we could define  $k_2$  to be  $k_3(\text{number of conserved individuals}) + k_4(\text{length of overlapping region}) + k_5((\text{number of ELOH})/(\text{number of CLOH}))$ . The parameters must be tuned to the input data. For example, association tests will tune the parameter to favor putative deletions with many conserved individuals. We suspect that this problem is NP-complete for general  $N$ . In the case of the Multiple Sclerosis dataset, the matrix  $N$  contains small overlapping putative deletions and over 95% of  $N$  is non-putative deletions, that is,  $N$  is very sparse.

**Algorithm 1.** We start by giving an exact exponential algorithm which minimizes the objective function. Let  $x_i$  denote a set of overlapping putative deletions. For sparse  $N$  we can reduce the minimization function from  $\min_N$  to  $\min_{x_1..x_s}$  where  $x_1..x_s \in N$  and  $\{x_1..x_s\} \subseteq N$ . Since any particular putative deletion is defined by the ELOH sites, we can enumerate all feasible non-empty sets of ELOH sites for all  $x_i$ . Computing this for all putative deletions demands work proportional to  $\sum_{i=1}^s B(e_i)$  where  $e_i$  is the number of ELOH sites in  $x_i$  and  $B$  is the Bell number. In practice, we found that  $e_i$  is bounded by a small constant but this complexity is still unreasonable for most  $e_i$ .

**Algorithm 2.** For practical purposes, we've developed a greedy algorithm for cases where the exact exponential algorithm is unreasonable (Fig. 5). For each  $x_i \in N$ , the algorithm selects the component with the maximum *trio sharing*,

	SNP Sites											
Trio 1	1	0	0	1	1	0	0	X	0	0	X	X
Trio 2	0	X	1	0	1	1	X	0	0	X	1	X
Trio 3	X	X	1	0	1	0	0	0	0	0	0	X
Trio 1	1	0	0	1	1	0	0	X	0	0	X	X
Trio 2	0	X	1	0	1	1	X	0	0	X	1	X
Trio 3	X	X	1	0	1	0	0	0	0	0	0	X
Trio 1	1	0	0	1	1	0	0	X	0	0	X	X
Trio 2	0	X	1	0	1	1	X	0	0	X	1	X
Trio 3	X	X	1	0	1	0	0	0	0	0	0	X

**Fig. 5.** A visual depiction of the greedy algorithm for finding putative deletions (consistencies with particular parents are omitted for simplicity). The red rectangles denote trio SNP sites which have not been called yet. The blue rectangle denotes a called inherited deletion haplotype. A green rectangle denotes a genotype error call. First, the algorithm finds the component (a clique in  $G(V,E)$ ) with the maximum trio sharing: SNP sites 3-6. It checks if the score of this component and either calls it an inherited deletion or a set of genotyping errors (in this case the former). The intervals are updated by remove vertices and edges from the overlap graph and the algorithm continues. Both remaining components consisting of SNP sites 1 and 11 are both of size 1. These will most likely be called genotyping errors.

that is, the possibly overlapping putative deletions that include the most ELOH sites. Because every two ELOH sites in an inherited deletion must be pairwise compatible, this component is a clique. To find the maximum clique, we construct an overlap graph  $G(V, E)$  where  $h_i \in V$  if  $h_i$  is an ELOH in a putative deletion in this interval and  $(h_i, h_j) \in E$  if  $h_i$  and  $h_j$  are compatible. Identifying the maximum clique in this graph is NP complete. We therefore find maximum cliques using a greedy approach that iterates over a queue containing the compatible vertices, selecting the highest degree node  $v_m$  and adding it to the potential clique set if and only there is an edge between  $v_m$  and each vertex in the clique. At the end of this process, the algorithm calls the site(s) a deletion haplotype or genotyping error according to the objective function, clears the set, and continues until all vertices in the queue are processed.

### 3.4 Experimental Results on Simulated Data

We tested the algorithm using the same simulated phasing dataset. To simulate and score an error-prone GWAS dataset containing an LOH, we define six parameters, two metrics, and generate only one deletion in the genotype matrix (Table 2). We randomly select a set of trios and an interval in the simulated haplotype matrix to contain the generated deletion. After the site is selected, we place ELOH sites on the SNPs according to some probability (assumed independent for each SNP in the interval).

**Table 2.** Six tunable parameters and two scoring metrics for testing of the LOH algorithm

Probability of Error per Site	For all SNP-trio pairs, we add a Mendelian error according to this probability (assumed independent for each site).
Interval Length	The exact length of the generated deletion.
Trios in Deletion	The exact number of trios sharing the generated deletion.
Probability of ELOH in Interval	The probability a SNP is an ELOH site within the generated deletion interval.
Coefficient of Genotype Error Call	The objective function cost for calling an ELOH site a genotyping error (parameter $k_1$ in our objective function)
Coefficient of Inherited Deletion Call	The objective function cost for calling a set of ELOH sites an inherited deletion (parameter $k_2$ in our objective function)
True Positive	There is one interval that contains the inherited deletion, thus a true positive corresponds to correctly identifying an inherited deletion in this region.
False Positive	We have a false positive if we identify an inherited deletion in a region disjoint from the generated deletion's region.

**Table 3.** We tested out algorithm using the six tunable parameters as defined in Table 2. Each configuration was run with a coefficient of genotyping error of 2.

Param Set	Site Error Prob.	Interval Length	Trios in Deletion	Prob. of ELOH	Coeff. of Deletion	True Positive	False Positive	Runs
1	0.0001	5	5	0.75	11	1000	0	1000
2	0.0001	2	5	1	11	1000	0	1000
3	0.0001	2	5	1	11	1000	0	1000
4	0.0001	9	3	0.75	11	1000	0	1000
5	0.0001	7	3	0.50	15	58	0	100
6	0.00333	9	3	0.75	15	100	38888	100

Although our LOH model is quite simplistic, we do observe promising results. Our algorithm is sensitive to inherited deletions that are very short but shared among many people and also sensitive to inherited deletions that are longer and shared by few people.

In general, the algorithm is accurate when the coefficient of deletion call and genotype error call are tuned well (Table 3 – parameter sets 1-4). For a dataset with low genotyping error rate ( $\sim 0.0001$  site error probability), the coefficient of deletion call can be set low; if it is set too high, a true inherited deletion may be incorrectly called a genotyping error, possibly missing an associative

LOH (Table 3 – parameter set 5). A similar caveat pertains to datasets with significant genotyping error rates (for instance, the MS dataset). A coefficient of deletion call that is too low can yield false positives (Table 3 – parameter set 6). Finding appropriate tuning mechanisms for the two coefficients to maximize algorithm specificity and sensitivity will be the subject of future work.

## 4 Conclusion and Future Work

We have shown that long range phasing using Clark consistency graphs is practical for very large datasets and the accuracy of the algorithm improves rapidly with the size of the dataset. We have also given an algorithm that removes most Mendelian inconsistencies and distinguishes between genotyping errors and deletion events which can be factored into the phasing algorithm when applied to GWAS data. Future work includes applying probabilistic models to both algorithms to score tract sharings and putative deletions more appropriately.

All algorithms are available via sending a request to the corresponding authors.

## Acknowledgments

Thanks to the International Multiple Sclerosis Genetics Consortium for sharing the Multiple Sclerosis GWAS dataset.

## References

- [1] Altshuler, D., Daly, M.J., Lander, E.S.: Genetic mapping in human disease. *Science* 322(5903), 881–888 (2008)
- [2] Browning, B.L., Browning, S.R.: A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American journal of human genetics* 84(2), 210–223 (2009)
- [3] Clark, A.G.: Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol.* 7(2), 111–122 (1990)
- [4] The International Multiple Sclerosis Genetics Consortium: Risk alleles for multiple sclerosis identified by a genomewide study. *N. Engl. J. Med.* 357(9), 851–862 (2007)
- [5] Gudbjartsson, D.F., Bragi Walters, G., Thorleifsson, G., Stefansson, H., Halldórsson, B.V., et al.: Many sequence variants affecting diversity of adult human height. *Nat. Genet.* 40(5), 609–615 (2008)
- [6] Halldórsson, B.V., Bafna, V., Edwards, N., Yooseph, S., Istrail, S.: A survey of computational methods for determining haplotypes (2004)
- [7] Howie, B.N., Donnelly, P., Marchini, J.: A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet.* 5(6), e1000529 (2009)
- [8] Hudson, R.R.: Generating samples under a wright-fisher neutral model of genetic variation. *Bioinformatics* 18(2), 337–338 (2002)
- [9] Istrail, S.: The haplotype phasing problem. In: *Symposium in Honor of Mike Waterman’s 60th Birthday* (2002)

- [10] Kong, A., Masson, G., Frigge, M.L., et al.: Detection of sharing by descent, long-range phasing and haplotype imputation. *Nat. Genet.* 40(9), 1068–1075 (2008)
- [11] McCarrroll, S.A., Kuruville, F.G., Korn, J.M., Cawley, S., et al.: Integrated detection and population-genetic analysis of snps and copy number variation. *Nat. Genet.* 40(10), 1166–1174 (2008)
- [12] Minichiello, M.J., Durbin, R.: Mapping trait loci by use of inferred ancestral recombination graphs 79(5), 910–922 (2006)
- [13] Rivadeneira, F., Styrkarsdottir, U., Estrada, K., Halldorsson, B.: Twenty loci associated with bone mineral density identified by large-scale meta-analysis of genome-wide association datasets. In: *Bone*, June 2009, vol. 44, pp. 230–231. Elsevier Science, Amsterdam (2009)
- [14] Scheet, P., Stephens, M.: A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase 78(4), 629–644 (2006)
- [15] Sharan, R., Halldórsson, B.V., Istrail, S.: Islands of tractability for parsimony haplotyping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3(3), 303–311 (2006)
- [16] Siva, N.: 1000 genomes project. *Nature biotechnology* 26(3), 256 (2008)
- [17] Stefansson, H., Rujescu, D., Cichon, S., Pietilainen, O.P.H., et al.: Large recurrent microdeletions associated with schizophrenia. *Nature* 455(7210), 232–236 (2008)
- [18] Stephens, M., Smith, N.J., Donnelly, P.: A new statistical method for haplotype reconstruction from population data 68(4), 978–989 (2001)
- [19] Styrkarsdottir, U., Halldorsson, B.V., Gretarsdottir, S., Gudbjartsson, D.F., Bragi Walters, G., et al.: Multiple Genetic Loci for Bone Mineral Density and Fractures. *N. Engl. J. Med.* 358(22), 2355–2365 (2008)

# A New Algorithm for Improving the Resolution of Cryo-EM Density Maps

Michael Hirsch<sup>1</sup>, Bernhard Schölkopf<sup>1</sup>, and Michael Habeck<sup>1,2</sup>

<sup>1</sup> Max-Planck Institute for Biological Cybernetics, Tübingen, Germany

<sup>2</sup> Max-Planck Institute for Developmental Biology, Tübingen, Germany

{firstname.lastname}@tuebingen.mpg.de

<http://www.kyb.mpg.de/bs/>

**Abstract.** Cryo-electron microscopy (cryo-EM) plays an increasingly prominent role in structure elucidation of macromolecular assemblies. Advances in experimental instrumentation and computational power have spawned numerous cryo-EM studies of large biomolecular complexes resulting in the reconstruction of three-dimensional density maps at intermediate and low resolution. In this resolution range, identification and interpretation of structural elements and modeling of biomolecular structure with atomic detail becomes problematic. In this paper, we present a novel algorithm that enhances the resolution of intermediate- and low-resolution density maps. Our underlying assumption is to model the low-resolution density map as a blurred and possibly noise-corrupted version of an unknown high-resolution map that we seek to recover by deconvolution. By exploiting the nonnegativity of both the high-resolution map and blur kernel we derive multiplicative updates reminiscent of those used in nonnegative matrix factorization. Our framework allows for easy incorporation of additional prior knowledge such as smoothness and sparseness, on both the sharpened density map and the blur kernel. A probabilistic formulation enables us to derive updates for the hyperparameters, therefore our approach has no parameter that needs adjustment. We apply the algorithm to simulated three-dimensional electron microscopic data. We show that our method provides better resolved density maps when compared with B-factor sharpening, especially in the presence of noise. Moreover, our method can use additional information provided by homologous structures, which helps to improve the resolution even further.

## 1 Introduction

Cryo-electron microscopy (cryo-EM) and low-resolution X-ray crystallography are emerging experimental techniques to elucidate the three-dimensional structure of large biomolecular complexes [1,2,3,4]. A major drawback common to these methods is that the reconstructed density maps are only of intermediate or low resolution, typically in the nanometer range. In this resolution range, it becomes difficult to interpret the density maps unambiguously and to fit atomic models. A method to improve the quality of electron density maps has therefore the potential to broaden the scope of cryo-EM and low-resolution crystallography.



B-factor sharpening [5,6,7] is often advocated as a method for improving the resolution of density maps. The method operates in the frequency domain and applies a negative B-factor to the Fourier coefficients of the density map. This has the effect that high-frequency components encoding high-resolution features are amplified. B-factor sharpening has several limitations: First, the underlying model of the PSF is an isotropic Gaussian whose width is determined by the magnitude of the overall B-factor (the Fourier transform of a Gaussian is a Gaussian with inverted width). This assumption may be inappropriate for anisotropic data such as 2D crystals. Second, the method suffers from amplification of noise: Noise in density maps contributes high-frequency components, which are weighted up when applying a negative B-factor. Third, it is not possible to incorporate prior knowledge to regularize the recovered high-resolution density map. For example, the B-factor sharpened density map is not guaranteed to be nonnegative.

In this article, we present a novel algorithm to sharpen electron density maps. The algorithm remedies some of the shortcomings of thermal factor sharpening. The underlying assumption is that low- to intermediate-resolution density maps can be viewed as distorted or “blurred” versions of high-resolution maps. Mathematically, this blurring process is modeled as a convolution

$$y = f * x \tag{1}$$

where  $y$  denotes the observed blurry and noisy low-resolution map,  $x$  the true high-resolution map,  $f$  the linear shift-invariant blur kernel or point spread function (PSF) and  $*$  the linear convolution operator.

We propose a blind deconvolution method (BD) to sharpen electron density maps. BD aims to invert the blurring process and thereby recover the high-resolution map without any knowledge on the degradation or blur kernel. It does so by estimating the sharpened density map and the PSF simultaneously. In this paper, we are interested in BD algorithms that do not assume a particular structural model and that are in this sense parameter-free. The recovered high-resolution map will be useful for density map interpretation and model fitting.

Blind deconvolution is a severely ill-posed problem because there exists an infinite number of solutions and small perturbations in the data lead to large distortions in the estimated true map. The ill-posedness may be alleviated by confining the set of admissible maps to those which are physically plausible through the introduction of additional constraints. One such constraint is that electron density maps are inherently nonnegative. We show that nonnegative blind deconvolution (NNBD) can be cast into a set of coupled quadratic programs that are solved using the multiplicative updates proposed in [8]. No learning rate has to be adjusted and convergence of the updates is guaranteed. By iterating between an update step for  $x$  and  $f$ , we obtain an efficient BD algorithm that allows for straightforward incorporation of prior knowledge such as sparseness and smoothness of the true map and/or the PSF.

Blind deconvolution is a valuable tool in many image and signal processing applications such as computational photography, astronomy, microscopy, and medical imaging and thus has been treated in numerous publications. Many

blind deconvolution algorithms have been proposed in various fields of research, for an overview confer [9,10,11,12]. However, to our knowledge it has never been proposed in the field of cryo-EM.

## 2 Blind Deconvolution by Nonnegative Quadratic Programming

Our generative model underlying the image formation process is

$$y \approx f * x$$

where the degraded map  $y$ , the PSF  $f$  and the true map  $x$  are  $n$ -dimensional<sup>1</sup>. Assuming additive Gaussian noise with zero mean and variance  $\tau^{-1}$ , the likelihood of observing  $y$  is given by

$$p(y|f, x, \tau) = Z(\tau)^{-1} \exp\left\{-\frac{\tau}{2} \|y - f * x\|^2\right\}$$

where  $\|\cdot\|$  denotes the  $L_2$ -norm and  $Z$  the normalizing partition function, which depends only on the precision  $\tau$ . As a prior, we constrain  $f$  and  $x$  to be of finite size and to lie in the nonnegative orthant:  $p(x) \propto \chi(x \geq 0)$  and  $p(f) \propto \chi(f \geq 0)$  where  $\chi$  is the indicator function. Computation of the maximum a posteriori (MAP) estimate of  $f$  and  $x$  is equivalent to the nonnegatively constrained problem of minimizing the negative log-likelihood viewed as a function of the unknown parameters  $f$  and  $x$ :

$$\min_{f \geq 0, x \geq 0} L(f, x) = \frac{1}{2} \|y - f * x\|^2. \quad (2)$$

Here, the negative log-likelihood  $L$  is expressed in units of  $\tau$  and constants independent of  $f$  and  $x$  have been dropped. Because of the interdependence of  $f$  and  $x$  through the convolution, optimization problem (2) is non-convex and a globally optimal solution cannot be found efficiently. Fortunately, the objective function  $L(f, x)$  is sufficiently well-behaved as it is convex in each variable separately if the other is held fixed. This observation suggests a simple alternating descent scheme: instead of minimizing (2) directly we iteratively solve the minimization problems  $\min_{f \geq 0} L(f)$  and  $\min_{x \geq 0} L(x)$ , where  $L(f)$ ,  $L(x)$  denotes  $L(f, x)$  for fixed  $f$ ,  $x$ , respectively. If we can ensure descent in each step, we will obtain a sequence of estimates  $\{f^{(k)}, x^{(k)}\}$  that never increase the objective  $L(f, x)$ . Due to the symmetry of the convolution operation,  $f * x = x * f$ , we can restrict our exposition to the optimization of  $x$ ; equivalent results will hold for  $f$ .

<sup>1</sup> The convolution is assumed to be non-circular and its value is taken only on its valid part, i.e. in the one-dimensional case, if  $x \in \mathbb{R}^n$  and  $f \in \mathbb{R}^m$ , then  $y$  is an element of  $\mathbb{R}^{n-m+1}$ . For discretized signals,  $*$  reads  $(f * x)_n = \sum_{i \in \text{supp}(f)} f_i x_{n-i}$  where  $\text{supp}(f)$  denotes the support of  $f$ .

Because convolution is a bilinear operation, the problem of optimizing  $x$  can be written in matrix notation:

$$\min_{x \geq 0} L(x) = \frac{1}{2} \|y - f * x\|^2 = \frac{1}{2} x^T F^T F x - y^T F x + \frac{1}{2} y^T y \tag{3}$$

where in this formulation  $y$ ,  $x$  and  $f$  are zero-padded vectors stacked in lexicographical order and  $F$  is a block-Toeplitz structured matrix. In the following we will use both notations interchangeably; the type of the involved quantities will be clear from the context. Minimizing (3) is equivalent to solving a quadratic program with nonnegativity constraint (NNQP)

$$\min_{x \geq 0} \frac{1}{2} x^T A x + b^T x \tag{4}$$

with  $A = F^T F$  and  $b = -F^T y$ . Recently, a novel algorithm for solving NNQPs based on multiplicative updates has been proposed [8]. In the derivation of the updates, only the positive semidefiniteness of  $A$  is required. In particular,  $A$  may have negative entries off-diagonal. The key idea is to decompose  $A$  into its positive and negative part, i.e.  $A = A^+ - A^-$  where  $A_{ij}^\pm = (|A_{ij}| \pm A_{ij})/2$ , and to construct an auxiliary function  $G(x, x')$  for the objective (2) such that  $\forall x, x' > 0$ :  $L(x) \leq G(x, x')$  and  $L(x') = G(x', x')$ . Because  $G(x, x')$  is an upper bound on  $L(x)$ , minimization with respect to  $x$  yields an estimate  $\hat{x} = \operatorname{argmin}_x G(x, x')$  which never increases the objective  $L(x')$ :

$$L(\hat{x}) \leq G(\hat{x}, x') \leq G(x', x') \leq L(x').$$

As shown in [8] a valid auxiliary function for (4) is given by

$$G(x, x') = \frac{1}{2} \sum_i \frac{(A^+ x')_i}{x'_i} x_i^2 - \sum_i (A^- x')_i x'_i \log \frac{x_i}{x'_i} + b^T x - \frac{1}{2} x'^T A^- x'. \tag{5}$$

Minimization of (5) with respect to its first argument yields the update:

$$x \leftarrow x \odot \frac{-b + \sqrt{b \odot b + 4(A^+ x) \odot (A^- x)}}{2A^+ x}. \tag{6}$$

The symbol  $\odot$  denotes voxel-wise multiplication, also division and square root are understood voxel-wise. For a nonnegative observed map  $y$  with  $A^+ = F^T F$ ,  $A^- = 0$  and  $b = -F^T y$ , update (6) reads

$$x \leftarrow x \odot \frac{F^T y}{F^T F x}. \tag{7}$$

Contrary to previous approaches to NNQP [13], no learning rate is involved that needs adjustment. Furthermore convergence to a global optimum is guaranteed. Note that as  $f * x$  approaches  $y$  the multiplicative factor in (7) tends to one. The update rules can be computed very efficiently using the Fast Fourier Transform [14] because

$$F x \equiv f * x = \mathcal{F}^{-1} \{ \mathcal{F}(f) \cdot \mathcal{F}(x) \}$$

and

$$F^T x \equiv f \star x = \mathcal{F}^{-1} \{ \mathcal{F}(f) \cdot \mathcal{F}(x) \}$$

**Algorithm 1.** Nonnegative Blind Deconvolution

---

**Input:** Degraded, blurry map  $y$   
**Output:** Sharp map  $x$ , blur kernel  $f$

Initialization of  $f$  and  $x$  with positive flat maps  
**while**  $\|y - f * x\|_F^2 > \epsilon$  **do**

$f \leftarrow x \odot \frac{f * y}{f * f * x}$
$x \leftarrow f \odot \frac{x * y}{x * x * f}$

**end**  
**return**

---

where  $\mathcal{F}$  denotes the discrete Fourier transform and  $\star$  the  $n$ -way correlation between  $f$  and  $x$ . Hence, we never have to compute matrices  $F$  and  $X$  explicitly. Because the objective is symmetric in  $x$  and  $f$ , we obtain an equivalent update for  $f$ :

$$f \leftarrow f \odot \frac{X^T y}{X^T X f}. \quad (8)$$

Coming back to our original problem, namely solving (2) jointly in  $x$  and  $f$ , we propose to iterate between update steps in  $x$  and  $f$ . Cycling between (7) and (8) ensures that both  $f$  and  $x$  will remain in the nonnegative orthant. Although multiplicative updates guarantee convergence to a global optimum in the case of NNQP, the proposed NNBD scheme only ensures convergence to a stationary point. Therefore, the solution might be sensitive to the initial values of  $x$  and  $f$ . In our experiments, however, initialization was never a problem: choosing flat maps for the initial  $x$  and  $f$  always led to good results. Algorithm (1) summarizes our NNBD approach.

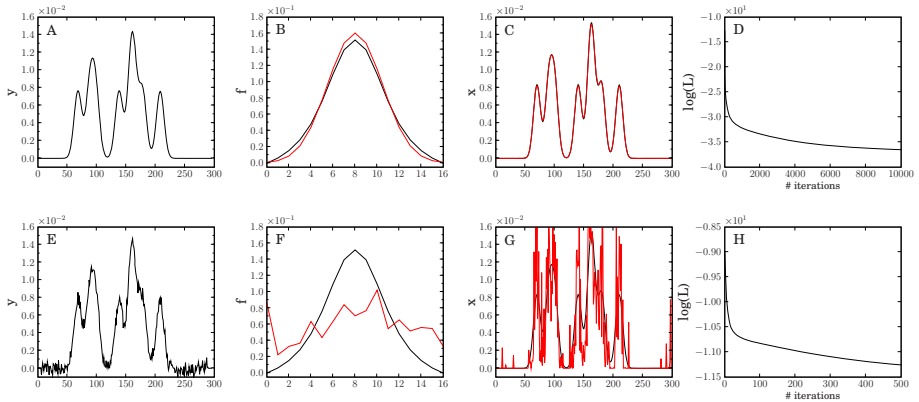
### 3 Incorporation of Prior Knowledge

In the absence of noise as well as in the case of high signal-to-noise ratios<sup>2</sup> (SNRs) our algorithm correctly decomposes a blurry observation into the true underlying map and the corresponding PSF<sup>3</sup>. Figure 1 shows a simulated one-dimensional toy example, where  $x$  is an equispaced sample of a Gaussian mixture model and  $f$  is chosen such that it is irreducible<sup>4</sup>. The estimated map  $\hat{x}$  and PSF  $\hat{f}$  are

<sup>2</sup> Here, we define the SNR of a signal as  $\text{SNR}(\text{dB}) = 10 \log_{10} \frac{\text{var}(x)}{\text{var}(y - x * f)}$ .

<sup>3</sup> Note that this is true only up to an overall scaling factor, because for each estimate  $\{\hat{f}, \hat{x}\}$  there exist infinitely many estimates  $\{\frac{1}{\lambda} \hat{f}, \lambda \hat{x}\}$  with  $\lambda \in \mathbb{R}^+$  that explain the observed data equally well. To rule these out, we fix the scale by normalizing  $f$ . In addition to this scale invariance, the solution is also shift-invariant. Usually this effect can be corrected only by means of further prior knowledge.

<sup>4</sup> A signal  $x$  is irreducible, if it cannot be decomposed into two or more nontrivial components  $\{x_1, x_2, \dots, x_n\}$  such that  $x = x_1 * x_2 * \dots * x_n$ . Note that if either  $f$  or  $x$  is reducible, NNBD becomes inherently ill-posed, because  $y = f * x$  cannot be decomposed unambiguously without employing additional prior knowledge.



**Fig. 1.** One-dimensional toy example. The top row shows the results of NNBD at SNR of 60 dB, the bottom row for SNR 20 dB. A, E: data  $y$  used in NNBD. B, F: true (black) and estimated (red) PSF  $f$ . C, G: true (black), NNBD (red) estimate of the true signal  $x$ . D, H: negative log-likelihood (on logarithmic scale).

close to the ground truth. However, Fig. 1 shows that low SNRs raise difficulties in the reconstruction process and lead to noise-fitting and unfavorable solutions.

To further constrain the space of admissible solutions, additional knowledge about the unknown map and the PSF has to be utilized. This knowledge will be represented by non-uniform prior distributions  $p(f|\theta)$  and  $p(x|\theta)$  on  $f$  and  $x$ , respectively, involving hyperparameters  $\theta$ . With  $p(\theta)$  denoting the prior of the hyperparameters, the joint posterior is proportional to:

$$p(x, f, \theta|y) \propto p(y|f, x, \theta) p(x|\theta) p(f|\theta) p(\theta). \quad (9)$$

In the following, we describe prior distributions that are compatible with the multiplicative updates for  $f$  and  $x$  derived in the previous section. Again, because of the symmetry of (2) in  $f$  and  $x$ , we will restrict ourselves to the incorporation of prior knowledge on the unknown map  $x$ .

Incorporating priors on  $x$  introduces additional terms in (3) that have to be taken into account in the computation of the MAP estimate. In the derivation of the multiplicative update rule (6), we minimized the auxiliary function (5) defining an upper bound on  $L(x)$ . A close look reveals that all priors whose negative logarithm comprises terms that are either linear, quadratic, or logarithmic in  $x$  can be incorporated into (5) and hence are compatible with the update (6). This includes the following priors:

- **Smoothness:** A desired property in many imaging applications is smoothness of the true map, which can be enforced by penalizing the norm of its gradient  $\|\nabla x\|$ . The corresponding prior is

$$p(x|\lambda) \propto \exp\left\{-\frac{\lambda}{2}\|\nabla x\|^2\right\} \quad (10)$$

Note that  $\|\nabla x\|^2$  can be rewritten as  $x^T \Delta x$  where  $\Delta x \equiv \nabla^T \nabla x = -\mathcal{L} * x$  is the negative Laplace operator, i.e. in the one-dimensional case  $\mathcal{L} = (1, -2, 1)$ .

- **Sparseness:** A further assumption commonly made is sparseness, which can be encoded in the exponential prior

$$p(x|\lambda) \propto \exp\left\{-\lambda \sum_i |x_i|\right\} = \exp\{-\lambda \mathbb{I}^T x\} \quad (11)$$

where the second equality holds for nonnegative maps.

- **Orthogonality:** In some applications, it is useful to introduce a voxel-wise nonnegative background  $z$ , which results in the model  $y = f * x + z$ . Such a background could, for example, account for the solvent in electron microscopic recordings or a homologous structure for model refinement (cf. section 4.1). Usually, the background should be uncorrelated with the reconstructed map which can be enforced by penalizing the overlap between  $x$  and  $z$ , i.e.

$$p(x|\theta) \propto \exp\{-\lambda z^T x\}. \quad (12)$$

We treat the background as a variable that we learn along with  $f$  and  $x$  using analogous multiplicative updates. In the following, we will refer to this regularization term as orthogonality constraint. Of course,  $z$  could be constant if such knowledge is available.

- **Entropy:** A reasonable assumption, especially for the form of the PSF, is that it exhibits a bump-like shape. This can be favored by using the entropic prior

$$p(x|\lambda) \propto \exp\left\{\lambda \sum_i \log x_i\right\}. \quad (13)$$

The Burg entropy  $\sum_i \log x_i$  is compliant with the auxiliary function  $G(x, x')$  and favors maximum entropy maps, i.e. constant maps. Entropy and sparseness/orthogonality can be combined into a single prior density: a voxel-wise Gamma distribution.

Table 1 summarizes the presented prior distributions and the required modifications in (6).

### 3.1 Estimation of Hyperparameters

An important aspect is the estimation of the unknown hyperparameters. Instead of resorting to heuristics or cross-validation, we use Bayesian inference to estimate the hyperparameters  $\theta$ . For all hyperpriors introduced in the previous section, the Gamma distribution  $G(\theta|\alpha, \beta)$  is a conjugate prior. The ideal approach to hyperparameter estimation would be to calculate their marginal posterior distribution

$$p(\theta|y) = \int_{f \geq 0} \int_{x \geq 0} p(x, f, \theta|y) \, df \, dx \quad (14)$$

and determine the mean or mode [15]. In our case, however, exact integration over  $f$  and  $x$  is infeasible. One would have to resort to computationally intensive

**Table 1.** Modifications for the incorporation of prior knowledge in the update of the true map.  $\Delta^+$  and  $\Delta^-$  refer to the decomposition of the negative Laplacian  $\Delta = \Delta^+ - \Delta^-$ .  $\text{diag}\{x\}$  is a diagonal matrix with entries  $x_i$ .

Prior	$A^+$	$A^-$	$b$
Smoothness	$F^T F + \Delta^+$	$\Delta^-$	$-F^T y$
Sparseness	$F^T F$	0	$-F^T y + \lambda \mathbb{I}$
Orthogonality	$F^T F$	0	$-F^T y + \lambda z$
Entropy	$F^T F$	$\lambda \text{diag}\{x\}^{-2}$	$-F^T y$

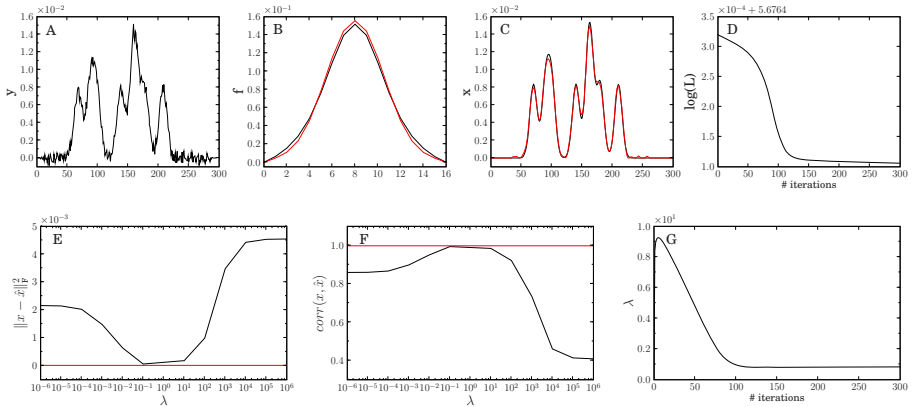
methods like Markov chain Monte Carlo or alternatives such as variational [16] or approximate inference [17]. Therefore we pursue the much simpler approach of computing the MAP estimate of the *joint* posterior, i.e.

$$\hat{\theta} = \text{argmin}_{\theta} p(\hat{f}, \hat{x}, \theta | y) \quad (15)$$

where  $\hat{f}$  and  $\hat{x}$  denote the MAP estimate of the PSF and the true map, respectively. Although it has been argued that this approximation is crude and neglects valuable information [12], the joint MAP approach led to good results in our experiments. The estimates for the hyperparameters  $\hat{\theta}$  can be derived by solving (15) directly. The shape parameters  $\alpha$  and  $\beta$  of the Gamma hyperprior are not estimated but set to fixed values  $\alpha = 1$  and  $\beta$  close to zero. According to [18] the sensitivity of the results on the shape parameters is negligible, which was confirmed by our experiments.

### 3.2 Discussion

Let us come back to the one-dimensional toy example at low SNR (cf. Fig. 1). Figures 2 A-D show how enforcing smoothness of the signal using prior (10) prevents unfavourable noise-fitting and effectively helps us to recover the original signal and the PSF from the blurred and noisy observation. We further investigated the estimation of the regularization parameter  $\lambda$ . We tested different fixed values for  $\lambda$  and compared the reconstruction error of and the correlation with the true signal when applying our hierarchical Bayes approach. Figures 2 E and F show that the Bayes procedure yields a minimal reconstruction error and a maximal correlation for a wide range of fixed  $\lambda$  values. The evolution of the regularization parameter (Fig. 2 G) reveals an important feature of our deconvolution algorithm. Starting at a small initial value, the regularization parameter increases rapidly within a few iterations after which it gradually converges to a smaller optimal value. This finding may justify the heuristic regularization scheme of Shan et al. [19], which seems to be crucial for the success of their BD algorithm on natural images [12]. Shan et al. propose to start the deconvolution with a large value of  $\lambda$  – a conservative choice that puts higher weight



**Fig. 2.** One-dimensional toy example. The top row shows the results of NNBD at a SNR of 20 dB. A: data  $y$  used in NNBD. B: true (black) and estimated (red) PSF  $f$ . C: true (black), NNBD (red) estimate of the true signal  $x$ . D: negative log-likelihood (on logarithmic scale). The bottom row shows the absolute deviation (E)/correlation coefficient (F) of the reconstructed signal  $\hat{x}$  from/and the true underlying signal  $x$  for fixed values of the regularization parameter  $\lambda$  (black) and in the case of NNBD with additional hyperparameter estimation (red) after 5000 iterations. G: Evolution of the hyperparameter  $\lambda$  with increasing number of iterations.

on the prior than on the data. As the deconvolution improves, the regularization parameter is decreased to put more and more weight on the data. This is similar to simulated or deterministic annealing which aims to avoid trapping in sub-optimal local minima. The advantage of our approach is that, contrary to Shan et al., we do not need to choose a schedule for adjusting  $\lambda$ . Rather our update procedure automatically balances the influence of the data versus the importance of the prior.

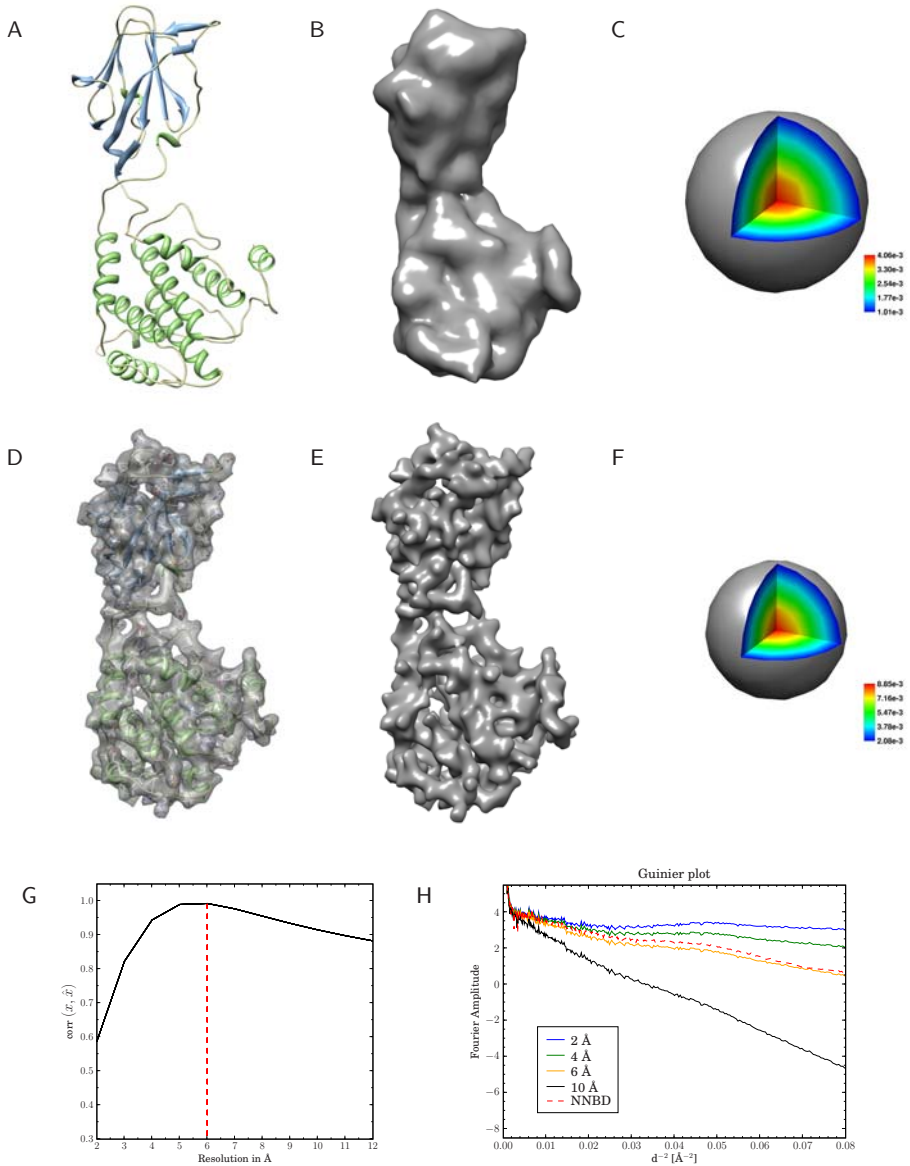
## 4 Applications

To evaluate the performance of our model and verify its validity we applied our algorithm to simulated three-dimensional density maps with a sampling of 1 Å/voxel. We used the program *pdb2mrc* from the EMAN software package [20] for density map simulation. First, we use nonnegative blind deconvolution to sharpen electron density maps. In the second application, we demonstrate the capabilities of our approach and the usefulness of the orthogonality prior by incorporating homologous structure information in the deconvolution.

### 4.1 Electron Density Maps of Proteins

For validation we used a monomer of the trimer of the bluetongue virus capsid protein VP7 (PDB ID: 2BTU) [21]. Figure 3 A shows the molecular structure,



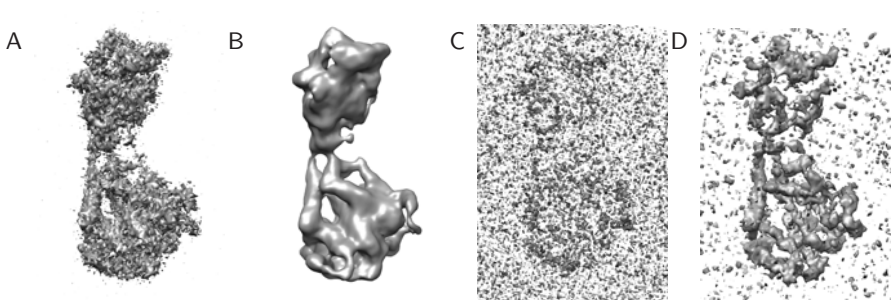


**Fig. 3.** NNBD for the electron density map of the monomer of the bluetongue virus outer shell coat protein VP7 (PDB ID: 2BTW): Top row: A: molecular structure, B: simulated density map at 10 Å, C: point spread function. Middle row: D: NNBD reconstruction with molecular structure fitted into it, E: NNBD reconstruction, F: estimated point spread function. Bottom row: G: correlation coefficient with simulated density maps at various resolutions, H: Guinier plot. See text for details.

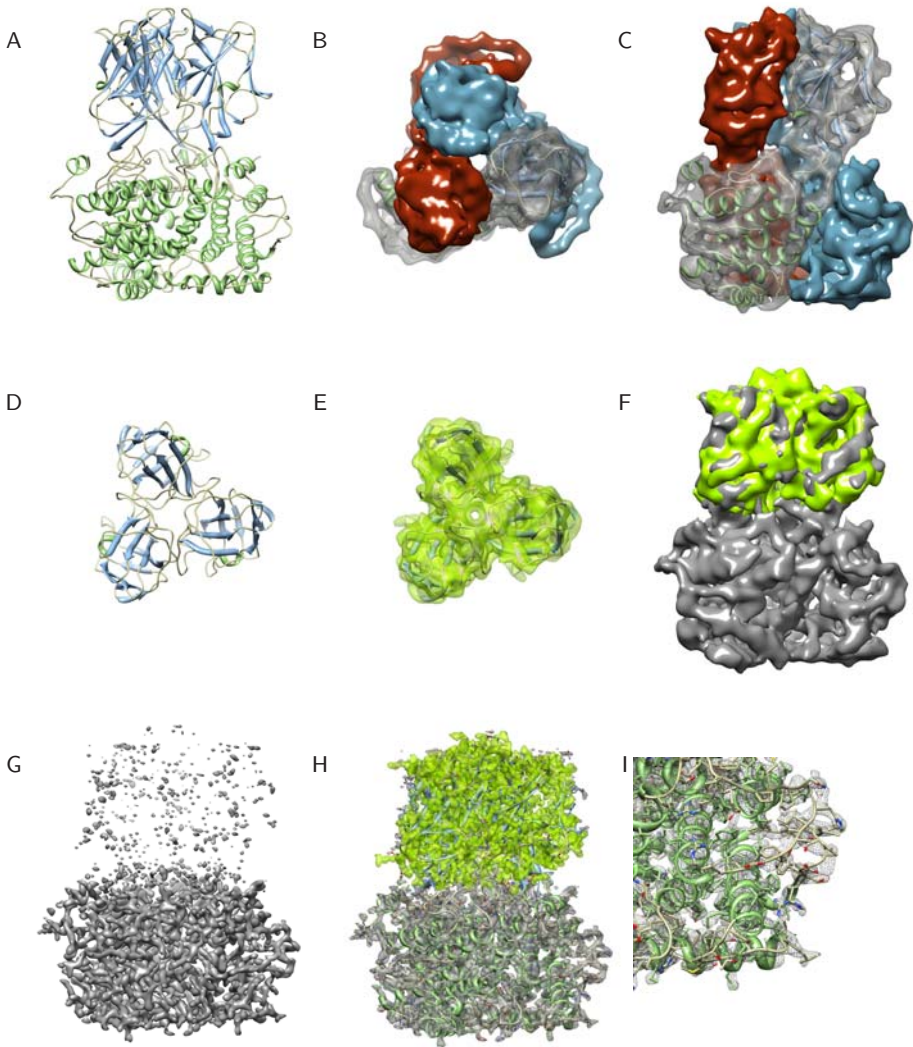
Fig. 3 B the simulated electron density map at 10 Å resolution and Fig. 3 C the corresponding PSF. Figures 3 D-F show the density map reconstructed with NNBD, the molecular structure fitted into it and the estimated PSF, respectively. The sharpened map reveals the nature of most secondary structure elements, whereas the original density map provides ambiguous secondary structure information. Also side chains become visible, which is important for modeling atomic details. To quantify the gain in resolution, we computed the correlation coefficient between the sharpened map and density maps simulated at higher resolutions. Figure 3 G shows that the correlation coefficient is highest for a density map at a resolution of 6 Å. Hence, our algorithm is able to sharpen the original map and to improve its resolution by almost a factor of two. Figures 3 C and F depict the true and estimated PSFs. The overall shape and functional form is determined correctly, however the estimated bandwidth appears to be smaller. This shrinkage of the PSF is largely due to the smoothness prior that downweights high-frequency components, which causes a loss of structural details but, at the same time, prevents amplification of noise. In this sense, underestimation of the bandwidth is conservative and should be viewed as a feature rather than a shortcoming.

Further insight is obtained by looking at the Guinier plot (Fig. 3 H) showing the radially averaged power spectrum against the squared resolution. In physical terms, the Guinier plot quantifies the map’s energy content at various spatial frequencies. Blurring has the effect that the Guinier plot drops off quite rapidly – convolution with a broad PSF acts as a low-pass filter that deletes all information above a certain cutoff frequency. The NNBD algorithm is able to recover high-frequency information to a large extent and lifts the Guinier curve above the curve of the simulated density map at a resolution of 6 Å (orange line in Figure 3 H).

To study the influence of noise, we corrupted the simulated density maps with Gaussian noise at different SNRs. We used the program *proc3d* from the EMAN software package [20] for noise corruption. Figure 4 A shows a noisy 10 Å-density



**Fig. 4.** NNBD for the electron density map of the monomer of the bluetongue virus outer shell coat protein VP7 (PDB ID: 2BTv): A: simulated density map at SNR of 6 dB at 10 Å resolution, B: NNBD reconstruction, C: result of embfactor, D: median-filtered result of embfactor

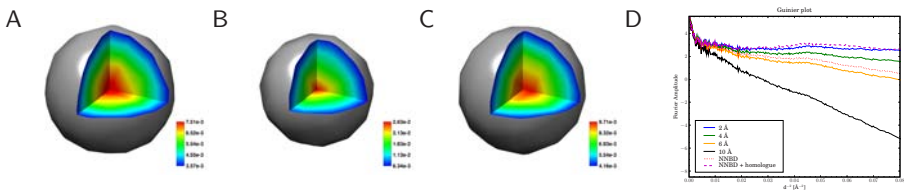


**Fig. 5.** NNBD for the electron density map of the bluetongue virus capsid protein (PDB ID: 2BTV) using additional structural information from a homologous fold. Top row: A: molecular structure of trimer 2BTV, B: top view of simulated density map of 2BTV at 8 Å resolution, C: sideview. Middle row: D: molecular structure of the African horse sickness virus capsid protein (PDB ID: 1AHS), E: simulated density map of 1AHS at 8 Å resolution, F: density map of 1AHS fitted into the map of 2BTV by FOLDHUNTER. Bottom row: G: NNBD of 2BTV without density map of homologous fold, H: molecular structure of 2BTV fitted into the density map, I: closeup view of H. See text for details.

map at a SNR of 6 dB. Figure 4 B shows the corresponding NNBD reconstruction using a smoothness prior (10). For comparison Fig. 4 C shows the density map sharpened with *embfactor* (7,6), the state-of-the-art method within the field.

## 4.2 Incorporating Homologous Structure Information

We now demonstrate how additional information from homologous structures can be incorporated to aid the deconvolution process and to detect secondary structure. We use the trimeric structure of the bluetongue virus capsid protein VP7 (PDB ID: 2BTV) as an example. Figures 5 A-C show the molecular structure, a top and side view of the simulated density of 2BTV at a resolution of 8 Å. The protein is made up of  $\beta$ -sheets and  $\alpha$ -helices in the upper and lower domains, respectively. The African horse sickness virus capsid protein (PDB ID: 1AHS) is a close structural homologue (RMSD: 1.4 Å) to the all-beta domain of 2BTV. Figures 5 D-F display the molecular structure, the simulated density at a resolution of 8 Å and the fit of 1AHS into 2BTV provided by FOLDHUNTER (22). In B-factor sharpening, information from homologous folds is used to compute the optimal B-factor for density sharpening. In our blind deconvolution approach, we model the observed density map as being composed of the homologous structure simulated at a higher resolution and the remainder density of 2BTV. The density of the homologous fold is held fixed, only the missing density and the PSF are estimated during the deconvolution. As initial PSF, we use a Gaussian at 6 Å resolution corresponding to the resolution difference between the high-resolution density of 1AHS at 2 Å and the experimental density. During reconstruction, we apply the orthogonality constraint (12) to enforce that the 1AHS density and the unexplained region of 2BTV do not overlap. The result of NNBD is shown in Figs. 5 G-I. As clearly visible in the closeup (Fig. 5 I), the sharpened density map reveals sidechains and information with almost atomic resolution. Figures 6 A-C compare the true PSF and the PSFs estimated by NNBD with and without homologous structure. As in the previous example, the width of the PSF is underestimated due to the smoothness prior. However, the additional structural information facilitates a more accurate estimation of



**Fig. 6.** Comparison of true PSF (A) and the PSFs estimated by NNBD without (B) and with homologous structure information (C). D: Guinier plot of reconstructed density maps with (magenta dashed line) and without homologous structure information (red dotted line). See text for details.

the PSF (Fig. 6C) and thereby allows the restoration of a high-resolution density map (Fig. 5I). The Guinier plot (Fig. 6D) illustrates the improved recovery of high-frequency information and the increase in resolution.

## 5 Summary

We propose a new method for improving the resolution of cryo-EM density maps by nonnegative blind deconvolution. We provide an iterative algorithm for learning simultaneously the sharpened density map and the blur kernel. We illustrate the generality of the proposed framework and show that the derived updates allow for easy incorporation of prior knowledge such as smoothness and sparseness. The updates are multiplicative and do not require the adjustment of a learning rate, as opposed to previously proposed gradient descent techniques. In addition, the updates ensure the nonnegativity of the sharp map and the PSF and guarantee convergence to a stationary point. A hierarchical Bayesian formulation also allows us to derive update rules for the hyperparameters, thus the method is fully parameter-free. The simplicity of the multiplicative updates allows for straightforward implementation. By employing the Fast Fourier Transform, we can reduce the computational complexity to large extent such that even medium and large sized problems (number of voxels  $> 10^7$ ) can be tackled efficiently. Computation time is typically in the order of minutes to hours for large density maps ( $> 400^3$ ) depending on the number of iterations one is willing to perform. Since our method allows the inspection of intermediate results, the user can decide when to stop either by visual inspection or by a user-set threshold of the monotonically decreasing cost function. We illustrate the performance and versatility of our algorithm by sharpening simulated electron density maps of the bluetongue virus capsid protein VP7 and by incorporating homologous structure information into the deconvolution process. We are currently applying our method to experimental density maps. Initial results confirm that NNDB is a flexible and generic tool to improve the resolution of electron density maps.

## References

1. Frank, J.: Single-particle imaging of macromolecules by cryo-electron microscopy. *Annu. Rev. Biophys Biomol. Struct.* 31, 303–319 (2002)
2. Orlova, E.V., Saibil, H.R.: Structure determination of macromolecular assemblies by single-particle analysis of cryo-electron micrographs. *Curr. Opin. Struct. Biol.* 14, 584–590 (2004)
3. Chiu, W., Baker, M.L., Jiang, W., Dougherty, M., Schmid, M.F.: Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure* 13, 363–372 (2005)
4. Brünger, A.T.: Low-resolution crystallography is coming of age. *Structure* 13, 171–172 (2005)
5. DeLaBarre, B., Brunger, A.T.: Considerations for the refinement of low-resolution crystal structures. *Acta Crystallographica D* 62, 923–932 (2006)

6. Rosenthal, P.B., Henderson, R.: Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *J. Mol. Biol.* 333, 721–745 (2003)
7. Fernández, J.J., Luque, D., Castón, J.R., Carrascosa, J.L.: Sharpening high resolution information in single particle electron cryomicroscopy. *J. Struct. Biol.* 164, 170–175 (2008)
8. Sha, F., Lin, Y., Saul, L.K., Lee, D.D.: Multiplicative Updates for Nonnegative Quadratic Programming. *Neural Comput.* 19(8), 2004–2031 (2007)
9. Kundur, D., Hatzinakos, D.: Blind Image Deconvolution. *IEEE Signal Processing Magazine* 13, 43–64 (1996)
10. Starck, J.L., Pantin, E., Murtagh, F.: Deconvolution in Astronomy: A Review. *The Publications of the Astronomical Society of the Pacific* 114, 1051–1069 (2002)
11. Sarder, P., Nehorai, A.: Deconvolution methods for 3-d fluorescence microscopy images. *IEEE Signal Processing Magazine* 23(3), 32–45 (2006)
12. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: *IEEE Conference on Computer Vision & Pattern Recognition* (2009)
13. Johnston, R.A., Connolly, T.J., Lane, R.G.: An improved method for deconvolving a positive image. *Optics Communications* 181, 267–278 (2000)
14. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes: The art of scientific computing*, 3rd edn. Cambridge University Press, Cambridge (2007)
15. Mackay, D.J.C.: Hyperparameters: Optimize, or integrate out? In: *Maximum Entropy and Bayesian Methods*, pp.43–59 (1996)
16. Molina, R., Mateos, J., Katsaggelos, A.K.: Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Transactions on Image Processing* 15, 3715–3727 (2006)
17. Lin, Y., Lee, D.D.: Bayesian regularization and nonnegative deconvolution for time delay estimation. In: *NIPS*, pp. 809–816 (2005)
18. Jin, B., Zou, J.: Augmented Tikhonov regularization. *Inverse Problems* 25(2), 025001 (2009)
19. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Transactions on Graphics, SIGGRAPH* (2008)
20. Ludtke, S.J., Baldwin, P.R., Chiu, W.: EMAN: semiautomated software for high-resolution single-particle reconstructions. *J. Struct. Biol.* 128, 82–97 (1999)
21. Grimes, J.M., Burroughs, J.N., Gouet, P., Diprose, J.M., Malby, R., Ziéntara, S., Mertens, P.P.C., Stuart, D.I.: The atomic structure of the bluetongue virus core. *Nature* 395, 470–478 (1998)
22. Jiang, W., Baker, M.L., Ludtke, S.J., Chiu, W.: Bridging the Information Gap: Computational Tools for Intermediate Resolution Structure Interpretation. *Journal of Molecular Biology* 308, 1033–1044 (2001)



# Towards Automated Structure-Based NMR Resonance Assignment

Richard Jang\*, Xin Gao\*, and Ming Li\*\*

David R. Cheriton School of Computer Science, University of Waterloo,  
Waterloo, Ontario, Canada N2L 6P7

**Abstract.** We propose a general framework for solving the structure-based NMR backbone resonance assignment problem. The core is a novel 0-1 integer programming model that can start from a complete or partial assignment, generate multiple assignments, and model not only the assignment of spins to residues, but also pairwise dependencies consisting of pairs of spins to pairs of residues. It is still a challenge for automated resonance assignment systems to perform the assignment directly from spectra without any manual intervention. To test the feasibility of this for structure-based assignment, we integrated our system with our automated peak picking and sequence-based resonance assignment system to obtain an assignment for the protein TM1112 with 91% recall and 99% precision without manual intervention. Since using a known structure has the potential to allow one to use only N-labeled NMR data and avoid the added expense of using C-labeled data, we work towards the goal of automated structure-based assignment using only such labeled data. Our system reduced the assignment error of Xiong-Pandurangan-Bailey-Kellogg's contact replacement (CR) method, which to our knowledge is the most error-tolerant method for this problem, by 5 folds on average. By using an iterative algorithm, our system has the added capability of using the NOESY data to correct assignment errors due to errors in predicting the amino acid and secondary structure type of each spin system. On a publicly available data set for Ubiquitin, where the type prediction accuracy is 83%, we achieved 91% assignment accuracy, compared to the 59% accuracy that was obtained without correcting for typing errors.

## 1 Introduction

Nuclear Magnetic Resonance (NMR)-based technologies are not only important for determining protein structure in solution [3, 7], but also studying protein-protein, protein-ligand interactions [31, 42], and identifying new drugs [35, 41]. However, presently, it can still take an experienced NMR spectroscopist weeks to months to process the data after the NMR spectra are collected. A key bottleneck step in the data processing is backbone resonance assignment, where the goal is to assign chemical shift values extracted from the spectra to the underlying backbone atoms. If the protein is examined under multiple experimental

---

\* The first two authors are Joint First Authors.

\*\* All correspondence should be addressed to mli@uwaterloo.ca

conditions, or different mutants of the protein are being studied, the assignment step needs to be repeated each time. Automated methods can accelerate this process especially if a similar 3D structure is already known, such as one obtained from a previous step. The use of a structure may also allow subsequent steps to use only N-labeled NMR data.

Traditional, sequence-based, resonance assignment methods depend mainly on the amino acid sequence and carbon connectivity information extracted from triple resonance experiments [1,4,6,10,17,19,21,22,23,24,27,28,30,32,36,43,46,50,51]. With the rate of new unique protein folds being discovered decreasing relative to the rate of protein structures being determined [33,34], one can expect that most proteins have homologs with a known protein structure. Analogous to molecular replacement in X-ray crystallography [12], the known structure can be used as a template to which the NMR experimental evidence is matched, as is done in various structure-based assignment methods [4,5,6,13,14,18,21,26,38,39,43,49,50].

The Nuclear Vector Replacement (NVR) approach [26,27] uses  $^{15}\text{N}$ -HSQC spectra,  $\text{H}^{\text{N}}\text{-}^{15}\text{N}$  residual dipolar couplings (RDC), sparse  $d_{\text{NN}}$  NOEs, amide exchange rates, and no triple resonance data for structure-based assignment. The problem was cast as a maximum bipartite matching problem, which they solved in polynomial time. Using close structural templates, they achieved an accuracy of over 99%. Their work was extended to handle more distant templates using normal mode analysis to obtain an ensemble of template structures [4]. Unlike NOEs, which stem from short-range interactions, RDCs can provide long-range orientation information. However, currently in NMR labs, RDC experiments are not as commonly used for backbone resonance assignment.

For assignment using 3D NOESY data, Xiong *et al.* developed a branch-and-bound algorithm [49], which they later improved to a randomized algorithm [50], which we shall refer to as the contact replacement (CR) method. The CR method was demonstrated to tolerate 1-2Å structural variation, 250-600% noise, and 10-40% missing contact edges. Although they mention that there exists methods with close to 90% average accuracy for predicting a spin system's amino acid class prior to an assignment, the CR method ignored such errors. The method achieved an assignment accuracy of above 80% in  $\alpha$ -helices, 70% in  $\beta$ -sheets, and 60% in loops. To our knowledge, it is the most error-tolerant structure-based assignment method in terms of the noise level. The data used consisted of only N-labeled spectra: 2D  $^{15}\text{N}$ -HSQC, 3D  $^{15}\text{N}$ -TOCSY-HSQC, 3D  $^{15}\text{N}$ -NOESY-HSQC, and  $^3\text{J}_{\text{HNH}\alpha}$  coupling constants derived from 3D HNHA. The problem was cast as a subgraph matching problem, where one graph consisted of the contacts in the known protein structure, and the other consisted of the NOESY cross peaks (NOEs) that connected spin system pairs. In general, the mapping of NOESY peaks to specific contacts is ambiguous due to experimental errors, missing peaks, and false peaks. Although the graph problem that was solved is NP-hard, Xiong *et al.* proved that under their noise model, the problem could be solved in polynomial time with high probability. In NOE-net [43], the problem



was also cast as a subgraph matching problem. Unlike the CR method, NOE<sub>net</sub> generates an ensemble of assignments containing all assignments compatible with the NMR data, and it requires only  ${}^1H^N-{}^1H^N$  NOEs. However, it requires unambiguous NOEs, such as those from 4D NOESY experiments, so the noise is less than that handled by the CR method.

In NMR studies, NMR spectra are often examined by visual inspection, where the cross peaks get picked by inspection, or by automatic methods but then checked by the scientist. The peaks get accumulated in a list of peaks, and this list can change during the study as errors and inconsistencies are discovered during the assignment step. Therefore, the peak picking and the resonance assignment steps are usually done together. We aim to build a system that automates this process without any manual intervention. To our knowledge, current structure-based methods are still semi-automated. The heart of our system is a novel and general 0-1 integer linear programming (ILP) model. We focus on structure-based assignment using only N-labeled NMR data because using a known structure has the potential to allow one to avoid the added expense of using C-labeled data. Nevertheless, the ILP model can be adapted to include carbon connectivity information from C-labeled data.

To test the feasibility of fully-automated structure-based assignment, we first build upon our earlier work on automated sequence-based resonance assignment, IPASS [1], which uses peak lists that are automatically picked from the spectra. These peak lists, which tend to be more noisy than manually picked peaks, are generated from our automated peak picking system, PICKY [2]. We used our method to refine the IPASS assignment to achieve an accuracy of 91% recall and 99% precision, an improvement over the input assignment, which had a recall value of 84% and precision of 97%. Recall is defined as  $C \div R$  and precision as  $C \div S$ , where  $C$  is the number of correct assignments,  $R$  is the number of residues that can be assigned, and  $S$  is the number of assignments made by the method. Typically, by accuracy we mean precision. Although the improvement is modest, we started directly from the spectra using systems that are completely automated.

For using only N-labeled data, automated and robust structure-based assignment is still a challenge. In comparison to the CR method, on 9 proteins from the data set used by the CR method, our method, on average, has 5 times fewer incorrect assignments. As a step towards robust assignment, we achieve further error tolerance by using the NOESY data to directly handle errors in predicting each spin system's amino acid and secondary structure type. This was tested on 5 proteins with typing errors introduced, and on a publicly available data set for Ubiquitin with a combined type prediction accuracy of 83% (both amino acid and secondary structure type correct). On Ubiquitin, we achieved an assignment accuracy of 91%, which is a large improvement over the 59% accuracy that was obtained without correcting for typing errors. Although we focused on resonance assignment using only N-labeled spectra, we also discuss generalizations of the ILP model to take into account other sources of data.

## 2 Methods

We use the graph representations from the CR method [50] to represent the template protein of known structure and the NMR data of the unknown target protein.

*Contact Graph:* Each residue in the template protein is represented by a vertex labelled with residue-related features. We use only amino acid and secondary structure type. Other possible features include predicted chemical shift values, back-computed RDCs [26], etc . . . . An edge is created between a pair of amino acids if there is a contact according to a given distance cutoff. Each edge is labelled by all pairs of directed proton-proton interaction types. We consider only two types of interactions,  $H^\alpha$  and  $H^N$ , and  $H^N$  and  $H^N$ . Since  $H^\alpha$  and  $H^N$  is not symmetric, the labels have a direction.

*Interaction Graph:* We define each spin system to consist of the chemical shifts of the backbone N,  $H^N$ ,  $H^\alpha$ , and the side chain protons. Each spin system is represented by a vertex labelled with spin system-related features. We use only the predicted amino acid and secondary structure type. Like the CR method, we use the side chain protons only in amino acid type prediction. Amino acid type predictions were obtained from the RESCUE software, version 1 [37]. RESCUE classifies each spin system into one of ten possible amino acid classes using proton chemical shifts. We used all classes with positive reliability score rather than the highest scoring class because this improved assignment accuracy. Secondary structure type predictions can be obtained from  ${}^3J_{HNH^\alpha}$  coupling constants [47]. Other possible features include experimental chemical shifts and RDC values. An edge is created between a pair of spin systems if there is at least one matching NOESY peak ( ${}^{15}\text{N}$ ,  $H^N$ ,  ${}^1\text{H}$ ), where the  ${}^{15}\text{N}$ ,  $H^N$  matches the backbone N,  $H^N$  chemical shift of one spin system and the  ${}^1\text{H}$  matches the backbone  $H^N$  or  $H^\alpha$  of the other spin system. Edges are labelled similarly to the contact graph with the addition of a match score for each NOESY peak. The match score is defined as  $\text{erfc}(\frac{|\Delta e|}{0.02 \times \sqrt{2}})$  as used in [50], where  $\text{erfc}$  is the complementary error function and  $|\Delta e|$  is the chemical shift difference between  ${}^1\text{H}$  and the matching  $H^N$  or  $H^\alpha$ . Edge labels are not limited to 3D  ${}^{15}\text{N}$  NOESY-HSQC data. If  ${}^1H^N$ - ${}^1H^N$  NOEs, as used in NOE-net, are available, a match score function that measures the chemical shift difference between the  $H^N$ s can be used.

To find the best match between the two graphs, we look for the common edge subgraph that maximizes the match score, subject to the constraint that the vertex and edge labels match. Finding the maximum common weighted edge subgraph (and also the maximum common node subgraph), in general, is NP-hard [40]. We use integer programming to do the maximization because it models the problem naturally as we will show. Our ILP formulation is similar to that for the maximum clique problem [8], to which subgraph matching can be reduced [40]. To solve the ILP model, we used the solver in the commercial optimization package ILOG CPLEX® version 9.130. Note that if we consider only vertex

matches, we get a maximum bipartite matching problem, which can be solved in polynomial time as in the NVR method.

## 2.1 0-1 Integer Programming Model

Define  $V_c, V_i$  to be the set of vertices in the contact graph and interaction graph, respectively. Define  $E_c, E_i$  to be the set of edges in the contact and interaction graph, respectively.

### Input Data

- $m(a, s, b, t)$  The edge match score between amino acids  $a, b \in V_c$  and spin systems  $s, t \in V_i$ , where  $a$  is matched with  $s$ , and  $b$  is matched with  $t$ . In our model, it is equal to the sum of the match scores of the NOESY peaks that match  $(s, t)$  and match an interaction type of  $(a, b)$ . The score is assumed to be non-negative.
- $m(a, s)$  The vertex match score between amino acid  $a$  and spin system  $s$ . The score is assumed to be non-negative
- $E_i(a, b)$  The set of edges in the interaction graph that match the edge  $(a, b) \in E_c$ . An edge  $(s, t) \in E_i$  matches edge  $(a, b)$  if the edge labels match while taking into account the direction of the interaction, and if either the label of vertex  $a$  matches that of vertex  $s$  and the label of vertex  $b$  matches that of vertex  $t$ , or  $a$  with  $t$  and  $b$  with  $s$ .
- $A$  The set of all matching  $(a, s)$ , where  $a \in V_c$  and  $s \in V_i$ , and there exists  $(a, b) \in E_c$  and  $(s, t) \in E_i$  such that  $(s, t)$  matches  $(a, b)$ .

### Decision Variables

- $X(a, s, b, t)$  A binary variable. It equals to 1 if spin system  $s$  is assigned to amino acid  $a$ , and spin system  $t$  is assigned to amino acid  $b$ ; and 0 otherwise. This variable represents an edge match between the graphs.  $X(b, t, a, s)$  is equivalent to  $X(a, s, b, t)$ . For the purpose of exposition, we use  $X(a, s, b, t)$  to denote either  $X(b, t, a, s)$  or  $X(a, s, b, t)$ , although the model contains only one such variable.
- $X(a, s)$  A binary variable. It equals to 1 if spin system  $s$  is assigned to the amino acid  $a$ ; and 0 otherwise. This variable represents a vertex match.

### Formulation

$$\max_X \left( \sum_{(a,s) \in A} m(a,s) \cdot X(a,s) + \sum_{(a,b) \in E_c} \sum_{\substack{(s,t) \in \\ E_i(a,b)}} m(a,s,b,t) \cdot X(a,s,b,t) \right) \quad (1)$$

subject to

$$\sum_s X(a, s) \leq 1 \quad \forall a \in V_c, \quad (2)$$

$$\sum_a X(a, s) \leq 1 \quad \forall s \in V_i, \quad (3)$$

$$\sum_{\substack{t \text{ s.t.} \\ (s, t) \in E_i(a, b)}} X(a, s, b, t) \leq X(a, s) \quad (4)$$

$$\forall (a, s) \in A, \forall (a, b) \in E_c,$$

$$X(a, s, b, t) \in \{0, 1\}, \quad (5)$$

$$X(a, s) \in \{0, 1\}. \quad (6)$$

**Discussion.** Equation (1), the objective function, expresses the total edge and vertex match score of the assignment. The first summation is over all vertices that are involved in at least one edge match. The second summation is over all edges that match. Unlike subgraph isomorphism, we look for edge matches only rather than non-matches. Non-matches are scored implicitly as described below. We generate only the variables involved in at least one edge match. We do not assign vertices that are isolated, unless the vertices can be unambiguously assigned, such as being the only ones with a particular type. Constraint (2) ensures that each amino acid is assigned to at most one spin system. Constraint (3) ensures that each spin system is assigned to at most one amino acid. Therefore, extra amino acids or spin systems can be unassigned, and missing amino acids or spin systems implicitly have a score of 0.

Constraint (4), in conjunction with (2) and (3), ensure that if  $X(a, s, b, t) = 1$ , then  $X(a, s) = 1$  and  $X(b, t) = 1$ . If  $X(a, s) = 1$  and  $X(b, t) = 1$ , the left hand side of (4) can be zero, so missing edges are allowed. However, edge match scores are always non-negative and we are maximizing the score. If a match exists, we are guaranteed that one edge match variable is set to 1. Note that (2) and (3) prevent the situation in (4) where  $X(a, s, b, t) = 1$  and  $X(a, u, b, v) = 1$ , or  $X(a, s, b, t) = 1$  and  $X(i, s, j, t) = 1$ , so each contact graph edge has at most one matching interaction graph edge that gets picked, and vice versa. Since the interaction graph tends to have more edges than the contact graph, extra edges can get unmatched. Since edge match scores are non-negative, missing edges implicitly have a score of 0, so a missing edge penalty for the scoring function is not necessary. To implicitly allow a negative missing edge penalty, all the edge match scores can be shifted by the penalty. The final two constraints ensure that the decision variables are binary. Note that the above formulation does not enforce that the common subgraph be connected, so contacts in different domains of the protein can get matched, while the parts in-between are unmatched.

## 2.2 ILP Model Generalizations

The ILP model can be adapted to accommodate different situations by setting, adding or removing variables, modifying their coefficients, and adding or removing constraints.

**Different Sources of Data.** Although we considered only chemical shift matches in the scoring function, the objective function of the ILP model can model any function that models the assignment of spins to residues and also the assignment of pairs of spins to pairs of residues. For C-labeled data, if there is carbon connectivity evidence that supports that spin systems  $s$  and  $t$  is associated with adjacent amino acids  $a_i$  and  $a_{i+1}$ , the value of  $m(a_i, s, a_{i+1}, t)$  can be increased. The variable  $X(a_i, s, a_{i+1}, t)$  can also be removed if there is insufficient connectivity and contact information.

For RDC data, once an alignment tensor has been estimated, back-computed RDCs can be computed and compared with the experimental values to yield a value for each  $m(a, s)$ . After running the ILP, the assignment information can be used to update the alignment tensor and  $m(a, s)$  terms. For  $^1H^N$ - $^1H^N$  NOEs, chemical shift matches can be encoded in the  $m(a, s, b, t)$  terms.

The coefficients  $m(a, s, b, t)$  did not use all the information in amino acids  $a$  and  $b$ . Different scores or weights can be used to account for matches to specific types of contacts in the template protein structure, such as long range  $\beta$ -sheet contacts and local  $H^\alpha$  and  $H^N$  contacts in  $\alpha$ -helices. The CR method focused on finding common Hamiltonian path fragments in the graphs to be matched. Similar to carbon connectivity, the score for matches to pairs of adjacent amino acids can be scaled up to emphasize the Hamiltonian path, so that the objective function contains a weighted version of the Hamiltonian path length. Alternatively, to enforce a maximum allowable number of missing edges along the path, we can add the constraint

$$\sum_{\substack{(a, b) \in E_c, (s, t) \in E_i(a, b) \\ |a - b| = 1}} X(a, s, b, t) \geq n - m \quad (7)$$

where the sum is over all spin system pair matches to adjacent amino acids.  $n$  is the number of amino acids minus one, and  $m$  is the maximum allowable number of missing edges along the path.

Note that if we remove the  $X(a, s, b, t)$  variables, and consider only the  $X(a, s)$  variables and use dummy vertices in the case that the size of  $V_c$  is not equal to  $V_i$ , we get a maximum bipartite matching problem. In this case, we can relax the constraint that the variables are integers because the constraint matrix becomes totally unimodular [9], so linear programming, which is not NP-hard, will give an integer optimal solution.

**Apriori Assignment Information.** ILP solvers can start from an initial solution to improve performance. This initial solution can even be a partial

assignment. If specific spin system-amino acid assignments are known, the corresponding vertex match variables can be fixed to 1. The ability to fix specific assignments and to start from an existing assignment allows for a semi-automated approach, where the returned assignment is examined and corrected manually. The ILP can then be rerun using the new information rather than starting from scratch.

**Multiple Solutions.** The maximum common subgraph is not necessarily unique, so there may be multiple best scoring assignments. The sequential algorithm, introduced by Greisdorfer et al. [16] and generalized to more than two solutions in [11], can be used to generate solutions that are within a certain percentage of the optimal solution and have maximum diversity as measured by a diversity measure, such as average pairwise hamming distance. The one tree algorithm can also be used [11]. Examining the variability of each amino acid's possible assignments among a set of optimal or near optimal assignments allows one to assess the assignment stability. The set of assignments can be used in consensus methods. For instance, the above ILP can be used to generate a consensus assignment by ignoring the  $X(a, s, b, t)$  variables and setting each  $m(a, s)$  to the number of times amino acid  $a$  got assigned to spin system  $s$ .

**NOE Assignment.** The current ILP model simplifies the assignment problem by using edge match variables, where each variable represents a match between an amino acid pair and spin system pair rather than between an atom and a spin. This leads to the problem where a given NOESY peak can explain more than one edge match. If we remove this problem and identify which NOESY peak corresponds to exactly which pair of contacting atoms, perhaps the accuracy of resonance assignment will improve. The ILP model can be modified to perform both resonance and NOE assignment simultaneously. However solving both problems increases the size of the model, so we leave it as future work. To enforce that each NOESY peak corresponds to at most one interaction, for each NOESY peak  $p$ , we have

$$\sum_{\substack{a, s, b, t, \\ (s, t) \text{ matches } p, \\ (s, t) \in E_i(a, b)}} X(a, s, b, t, p) \leq 1 \quad (8)$$

where we have defined a new binary variable  $X(a, s, b, t, p)$  corresponding to an edge match that is explained by NOESY peak  $p$ , where  $p$  matches spin systems  $s$  and  $t$ . To tie this variable to the other variables, we have,  $\forall X(a, s, b, t)$  where  $(s, t) \in E_i(a, b)$ ,

$$X(a, s, b, t) \leq \sum_{\substack{p, \\ (s, t) \text{ matches } p}} X(a, s, b, t, p) \quad (9)$$

$$k \cdot X(a, s, b, t) \geq \sum_{\substack{p, \\ (s, t) \text{ matches } p}} X(a, s, b, t, p) \quad (10)$$

where  $k$  is the number of NOESY peaks that match the spin system pair  $(s, t)$ . Constraint (9) ensures that if there is an edge match, the match is due to at least one NOESY peak. Constraint (10) ensures that if there are NOESY peaks explaining an edge match, the corresponding edge match variable will get selected. The  $m(a, s, b, t) \cdot X(a, s, b, t)$  terms in the objective function would then be replaced by terms of the form  $m(a, s, b, t, p) \cdot X(a, s, b, t, p)$ , where  $m(a, s, b, t, p)$  is the match score of NOESY peak  $p$  that matches  $(s, t)$  and matches an interaction type of  $(a, b)$ .

### 2.3 Spin System Type Prediction Errors

In the current ILP model, an edge match requires that the corresponding vertices match in amino acid and secondary structure type. If the type prediction for a spin system is incorrect, then it will get assigned to the wrong amino acid, and the correct spin system for that amino acid will also get incorrectly assigned. Assuming that the other assignments are correct, if the type matching requirement is then relaxed, we expect that the edge match scores for the incorrectly assigned spin systems will be greater when they are assigned to the correct amino acids. We do not, however, want to relax the type matching requirement for the correctly assigned spin systems. This forms the basis of our approach to handle type prediction errors as summarized in Fig. 1. The ILP model is first solved with the type matching requirement. Putative correct assignments are then identified, and then the ILP is resolved with these assignments fixed, while the type matching requirements are relaxed for the non-fixed spin systems.

To determine whether or not an assignment should be fixed, we examine the percentage of contacts matched involving each assigned amino acid. This percentage can be outputted as a confidence measure for each assignment. Due to erroneous assignments, a tight criteria for identifying fixed assignments may exclude correct assignments and result in a large problem size. For the initial criteria, we chose a 50% cutoff. Analogous to gradually decreasing the temperature in the simulated annealing optimization method [25], we used progressively tighter criteria. Once the ILP is resolved, the previously fixed assignments may no longer satisfy the criteria, while new assignments may satisfy it. Therefore, for a given criteria, we resolve the ILP until the fixed assignments do not change, or after a maximum number of iterations. We chose 50% because the majority of the missing edge percentages in our data are below 50% (Table 1). To tighten the criteria, we considered the requirement that a certain number of sequential neighboring contacts, nonlocal contacts between  $\beta$ -sheet amino acids, and local helix contacts ( $i \pm 5$ ) in the template protein structure be matched. We first required only one sequential neighbor and then later two (assignments for amino acids at the end points will not be fixed). Finally, we required that  $\beta$ -sheet amino acids have at least one  $\beta$ -sheet contact match, and that  $\alpha$ -helix amino acids have

at least one local contact match before and one local contact match after the residue. We did not attempt to optimize the set of criteria for fixing assignments as this is a modeling issue, and we wanted to show that our ILP model is flexible in modeling the problem.

For a given fixed assignment, ILP solvers can return a solution with score within  $N\%$  of the optimal solution, where we chose  $N$  to be 1%. If the fixed assignment is correct, then this solution will have score close to the global optimal solution. If not, other possible fixed assignments would need to be considered. We found that generating multiple solutions, improving each one, and then taking the best scoring one at the end produces a better final assignment. The generation of multiple solutions can be started at the initial ILP step or at subsequent ILP steps. In the latter case, previous assignments could be supplied to CPLEX as an initial feasible solution to speed up the optimization. Multiple solutions can also be generated from the final assignment by fixing assignments and then running the sequential or one tree algorithm. This allows the examination of the possible assignments for the non-fixed residues.

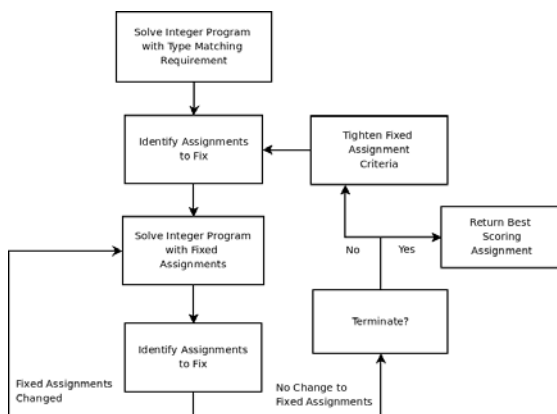


Fig. 1. Iterative Integer Programming with Fixed Assignments

### 3 Results

TM1112, an 89 residue protein from *Thermotoga maritima* [48], was provided by the Arrowsmith Lab at the University of Toronto. Accounting for 5 prolines, manual resonance assignment by the lab yielded 84 assigned residues. The protein has 17 residues in  $\alpha$ -helices, 58 in  $\beta$ -sheets, and 14 in loops. The assignment obtained without manual intervention from IPASS was refined using the X-ray structure PDB ID:1O5U and the spectra  $^{15}\text{N}$ -HSQC,  $^{15}\text{N}$ -edited NOESY, and HCCONH-TOCSY. HCCONH-TOCSY was used in place of 3D  $^{15}\text{N}$  TOCSY-HSQC because we did not have the latter. Since we did not have the latter, contacts from  $\text{H}_i^\alpha$  to  $\text{H}_j^N$  were represented by edges from  $\text{H}_i^\alpha$  to  $\text{H}_{j+1}^N$ . The



step that handles type prediction errors was also omitted because we did not have TOCSY-HSQC to do type prediction. Peak lists were obtained automatically from the spectra using PICKY. To compile the spin systems, chemical shift match tolerances of 0.5 ppm for N and 0.05 ppm for H were used. The IPASS assignment was fixed except where amino acids had greater than 70% of its contacts unmatched by the NOESY peaks of the assigned spin systems. We used an iterative approach of generating multiple assignments, taking the consensus, and discarding amino acids with no assigned spin system that occurred at least 50% of the time. This was repeated until no amino acids could be discarded. To handle the discarded amino acids, carbon connectivity information generated from IPASS was used to add any amino acids that could be unambiguously assigned.

Although we started with the IPASS assignment, this test is non-trivial because we used only a subset of the spectra used by IPASS, we could not do spin system type prediction, and the noise level (number of NOE edges per contact graph edge) was  $12\times$  at a distance cutoff of 4 Å. Nevertheless, by using contact information from the X-ray structure, we achieved an accuracy of 76 correct assignments out of 77 assigned amino acids, yielding 91% recall and 99% precision. This is a 5 residue improvement over the IPASS assignment, which achieved a recall value of 84% and precision of 97%. One wrong assignment made by IPASS was corrected. In general, the majority of the contacts of the unassigned residues were missing NOESY peaks. Although the improvement is modest, we started directly from the spectra using systems that are completely automated.

We then tested the performance of our method on the synthetic data set used by the CR method. It consisted of 9 proteins. The authors provided us with data that was simulated from the following NMR structures from the PDB: 1KA5, 1EGO, 1G6J, 1SGO, and 1YYC. The data for the other 4 proteins were simulated similar to their simulation method described in [49], where only one of the NMR models was used to generate the NOESY peaks. Although the simulated data was derived from one of the models in the PDB file, similar to the CR experiments, we tested the data using every model in the PDB file as the template structure, where the number of models per PDB file ranged from 10 to 32. The structural noise (in RMSD) of the models within each PDB file is given in Table 1, which summarizes the test set. To control noise, our method automatically increases the distance cutoff at 0.25 Å increments until the noise level is under 8. This gives an improvement over using a fixed 4 Å cutoff. We used the same distance cutoffs on the CR software.

Table 2 compares our method with the CR method, where the first row of each entry gives our results, while the row below gives the CR's. On 8 of the 9 proteins, our average accuracy on the entire protein is better. We achieved an average accuracy of 97.1%, whereas the CR method has 86.0% accuracy, resulting in 4.8 times fewer wrong assignments by our method. We also noticed that the ILP model significantly outperforms the CR method on both  $\beta$ -sheet and loop regions. This may be due to the fact that our method can maximize the score better as shown in column 4 of Table 2. In many instances, the score is higher than the score of the correct assignment, which indicates that maximizing

**Table 1.** Summary of the test set. From left to right: template structure, number of residues in the template (total/helix/sheet/loop); number of spin systems (total/helix/sheet/loop); number of prolines; noise level (number NOE edges per contact); percentage of contacts missing in the NMR data (total/helix/sheet/loop); average pairwise RMSD of the models in the template PDB file (total/helix/sheet/loop).

Template	No. Residues	No. Spin Sys	No. PRO	Noise (x)	Missing (%)	RMSD (Å)
1KA5	88/40/23/25	85/39/23/23	1	5.5/5.6/5.9/5.3	21/20/21/22	0.2/0.2/0.1/0.2
1EGO	85/40/19/26	81/40/19/22	3	5.6/5.4/5.8/6.3	22/22/26/19	1.6/1.4/0.9/2.3
1G6J	76/18/22/36	72/18/22/32	3	4.4/3.5/5.1/4.8	33/31/32/35	1.1/0.6/0.4/1.5
1SGO	139/46/28/65	136/46/28/61	3	5.5/4.7/4.0/7.4	41/38/49/40	10.9/7.3/5.5/14.1
1YYC	174/36/72/66	158/36/70/52	10	6.6/5.2/7.5/7.3	38/35/38/40	4.0/2.5/1.6/6.0
2NBT	66/-/16/50	60/-/16/48	5	3.4/-/3.6/3.3	36/-/22/40	3.4/-/1.7/3.8
1RYJ	70/9/27/34	67/9/27/31	2	3.1/2.0/3.1/3.8	28/33/29/25	1.5/1.0/0.9/1.9
2FB7	80/-/32/48	73/-/32/41	7	3.1/-/3.0/3.2	34/-/30/36	5.4/-/2.0/6.8
1P4W	87/66/-/21	82/65/-/17	3	5.5/5.3/-/6.7	31/28/-/40	1.1/0.7/-/1.9

contact matches alone may not necessarily give the correct assignment. For 2NBT, where 40% of loop contacts are missing, we did slightly worse, but the score is greater than the score of the correct assignment; similarly for helix residues in 1RYJ. In general, since amino acids in helices tend to have local contacts with nearby amino acids, in many of our tests, we observed that missing NOE edges and typing errors produced local errors in helices. For 1RYJ, the accuracy for helices using a ( $i \pm 2$ ) window, *i.e.*, allowing a spin system to be assigned within two residues away from the correct residue, was 100%.

Our program ran significantly faster. However, the CR program was written in Python, and ours was written in Java, and we used CPLEX. Both the CR and our program were run on our servers, consisting of Pentium 4 1.4Ghz, 4 GB RAM machines.

The CR software did not allow for the input of amino acid and secondary structure type predictions, so we could only perform the comparison assuming correct amino acid and correct secondary structure typing. Nevertheless, since perfect spin system typing cannot easily be achieved, we also tested our method on predicted spin system types. First we tested with only amino acid type prediction, and then we tested with both amino acid and secondary structure typing errors. For the 5 data sets received, we ran RESCUE Version 1 [37] on the experimental proton chemical shifts from the protein’s entry in the Biological Magnetic Resonance Bank (BMRB) [44]. Table 3 gives the results with amino acid type prediction. For comparison, we included the results of using type matching as strict constraints; that is, the result without using the iterative algorithm that tries to correct for typing errors. In general, type correction resulted in large improvements. For 1G6J, the amino acid typing accuracy is high, so the improvement is minimal. For 1YYC, the improvement is significant even though the typing accuracy is low. The accuracy, however, varied substantially depending on the model used as the template. Nevertheless, the template with the best score yielded an accuracy of 89.9%, which increases to 94.1% when considering

**Table 2.** Comparison between the ILP model and the Contact Replacement Method for correct amino acid and secondary structure typing. For each protein, the first row gives our results, while the second row gives the CR's. From left to right: template structure; average accuracy over all the models (total/helix/sheet/loop); accuracy ranges (total/helix/sheet/loop); number of times the assignment score was greater than, less than, or equal to the score of the correct assignment; the CPU time per model.

Template	Avg Acc. (%)	Acc. Range (%)	Times Score >, <, = Ref	CPU Time (sec)
1KA5	100/100/100/100	100/100/100	0, 0, 16	2
	94/100/76/100	98-93/91-74/100	0, 16, 0	804
1EGO	98/100/100/93	100-97/100/100/100-90	15, 0, 5	1
	96/96/100/93	100-92/100-90/100/100-79	4, 12, 4	708
1G6J	97/100/100/94	100-95/100/100/100-90	25, 2, 5	1
	91/100/ 87/88	97-89/100/100-86/100-85	0, 32, 0	756
1SGO	96/97/100/94	100-86/100-95/100/100-70	13, 3, 4	3
	80/95/95/62	88-71/100-87/100-86/76-45	0, 20, 0	4,302
1YYC	97/99/96/98	100-93/100/100-91/100-92	17, 0, 3	4
	72/92/62/72	76-67/100-89/69-53/79-64	0, 20, 0	5,292
2NBT	91/-/98/88	96-85/-/100-93/95-79	10, 0, 0	1
	92/-/95/90	100-88/-/100-88/96-82	1, 9, 0	2,328
1RYJ	97/98/96/96	97-94/100-88/96/96-93	20, 0, 0	1
	82/100/70/86	82-75/100/70/88-72	0, 20, 0	918
2FB7	96/-/97/96	100-91/-/100-93/100-90	7, 0, 3	1
	92/-/94/90	95-88/-/100-94/95-83	0, 10, 0	1,566
1P4W	99/100/-/97	100-97/100/-/100-88	4, 0, 16	3
	77/77/-/77	91-63/91-63/-/90-58	0, 20, 0	3,612
<b>Average</b>	97/99/99/96	-	-	2
	86/94/85/84	-	-	2254

an ( $i \pm 2$ ) window. This indicates that using multiple templates, such as those generated by normal mode analysis [4], may improve accuracy. In these tests, we used weaker criteria for fixing assignments. We did not require nonlocal  $\beta$ -beta sheet and local  $\alpha$ -helix contact matches.

Table 4 gives the results for both amino acid and secondary structure typing errors. The standard method for predicting secondary structures from  $^3J_{HNH\alpha}$  coupling constants [47] is similar to the following: if the coupling value is between 2.5 and 5.5, the spin system is predicted as helix. If the value is between 8 and 11.5, the spin system is predicted as  $\beta$ -sheet; otherwise, it is predicted as loop. From a test set of the following BMRB entries with accession numbers 4267, 4071, 2151, 4458, 4376, 4136, 4784, 4347, 4163, 4297, plus ubiquitin experimental values from the literature [45], we obtained an average typing accuracy of 60% with a range of 50-69%. This will likely be too low for resonance assignment, so we classified coupling constants into classes consisting of two secondary structure types, which dramatically increased the average accuracy at the cost of increased problem size. For values less than 6.5, we classify it as helix and loop; otherwise we classify it as  $\beta$ -sheet and loop. With this, we obtained an average accuracy of 92% with a range of 82-100%.

**Table 3.** Assignment accuracy for amino acid typing errors and correct secondary structure typing. From left to right: template structure; average accuracy for strict type matching; average accuracy for iterative error correction over all the models (total/helix/sheet/loop); accuracy ranges for iterative error correction (total/helix/sheet/loop); amino acid typing accuracy; number of times the assignment score was greater than, less than, or equal to the score of the correct assignment; the CPU time per model. Values in parenthesis give the accuracy within an  $i \pm 2$  window.

Template	Avg Acc		Range Acc	A.A. Typing	Times Score	CPU Time
	Strict (%)	Iter (%)	Iter (%)	Acc (%)	>, <, = Ref	
1KA5	86	100/100/100/100	100/100/100/100	89	0, 0, 16	30
1EGO	86	94/92(99)/100/94	100-91/100-87/100/100-90	90	15, 3, 2	22
1G6J	92	94/100/93/91	97-87/100/100-90/100-78	96	7, 25, 0	3
1SGO	82	92/90(100)/95/93	96-87/100-84/100-82/96-83	92	7, 13, 0	180
1YYC	59	77/86 (92)/81/66	94-68/100-58/100-52/90-5	79	0, 20, 0	504

For our tests, we introduced secondary structure class prediction errors yielding the typing accuracies in Table 4, which are slightly below 92%. In these tests, we used nonlocal  $\beta$ -beta sheet and local  $\alpha$ -helix contact matches for fixing assignments. For the convenience of time, we tested each target using only the first model in the template. The noise level and percentage of missing NOEs is similar to the average values in Table 1. From column 2 of Table 4, we see that low assignment accuracies can result if spin system type prediction errors are not handled, even if the type prediction accuracy is high. For 1KA5, the assignment accuracy did not change from the previous test. For 1EGO, the accuracy actually improved because of the tighter criteria for fixing assignments. The larger 1SGO struggled to maximize the score, but the accuracy is still much higher than without the iterative algorithm. For 1YYC, its large size combined with its low amino acid typing accuracy, produced poor quality fixed assignments, but there is still a large improvement over the case without the iterative algorithm.

For ubiquitin, we obtained  $^{15}\text{N}$  HSQC,  $^{15}\text{N}$  TOCSY-HSQC, and  $^{15}\text{N}$  NOESY-HSQC data from Richard Harris’s The Ubiquitin Resource Page [20]. We picked the peaks manually by inspecting the spectra with SPARKY [15]. Ubiquitin has 76 residues and 3 prolines. The noise level is 4.6 at 4 Å cutoff, and the missing edge percentage is 28.3%. HSQC peaks without an  $\text{H}^\alpha$  chemical shift were correctly filtered out as noise. For amino acid typing, RESCUE performed poorly, giving an accuracy of 68.6%. The errors appear to be due to missing peaks that are hidden by peak overlap. Using a higher resolution TOCSY spectrum may improve accuracy. We performed the typing manually using each type’s expected number of proton shifts and their expected range of values. Manual typing gave an accuracy of 90%, where the average number of possible amino acid types per spin system was 3.3 with a range of 1 to 8. We used the results of manual typing for assignment. We used experimental  $^3J_{\text{HNH}\alpha}$  coupling constants from the literature [45]. Eight spin systems did not have J-coupling values, so their predicted class included all three secondary structure types. The accuracy of secondary structure type prediction was 91%, yielding a combined typing accuracy of 83%. Model 1 from PDB 1D3Z was used as the template structure.

**Table 4.** Assignment accuracy for both amino acid and secondary structure typing errors. From left to right: template structure; accuracy for strict type matching; accuracy of the best scoring model for iterative error correction (total/helix/sheet/loop); amino acid typing accuracy; secondary structure typing accuracy; percentage difference in score of the best scoring assignment compared to the correct one (+ means score of our assignment was higher); the CPU time per model. Values in parenthesis gives the accuracy in a ( $i \pm 2$ ) window.

Template	Acc Strict	Acc Best Score	A.A. Typing	S.S. Typing	Diff Ref	CPU Time
	(%)	Iter (%)	Acc (%)	Acc (%)	Score (%)	
1KA5	72	100/100/100/100	89	91	0	4 hr
1EGO	65	97/95 (100)/100/100	90	85	-1.5%	1 h
1SGO	63	88/82 (91)/96/88	92	87	-3.0%	10.5 hr
1G6J	75	91/100/86/90	96	90	+0.5%	32 min
1YYC	40	70/91/71/53	79	91	-3.1%	46 hr

The template structure was not derived from the NMR data. An NMR model was used to test the case of using results from previous NMR studies. The best scoring assignment had accuracy 87.1%, with 64.3% on  $\alpha$ -helix (85.7% with  $i \pm 2$  window), 95.7% on  $\beta$ -sheet, and 90.0% on loops. Although the accuracy for helix residues is low, many of the errors are due to a +1 assignment position error due to the HSQC peak of a nearby amino acid not being present in the NMR data. We also obtained a consensus assignment by generating 10 solutions from the best scoring assignment with fixed assignments meeting the secondary structure contact matching criteria. Consensus gave an accuracy of 91% with 78% for helices (92%  $i \pm 2$ ) and the other types unchanged. Without the iterative algorithm, the accuracy is 59%.

## 4 Discussion

Local assignment errors in helices show the limitations of using only backbone proton contact information. Since our ILP model can accommodate different sources of information, it is of interest to test the relative contribution of each source to assignment. Our attempt at robust structure-based assignment using only N-labeled NMR data also reveals the challenges that impede complete automation. Amino acid type prediction from the unassigned chemical shifts of the side chain protons is impeded by missing and artifact TOCSY peaks, incorrect assignment of TOCSY peaks to their corresponding HSQC peak, and incorrect assignment of proton chemical shifts to their proton type ( $H^\beta$ ,  $H^\gamma$ , ...). Type prediction is further impeded by an increase in protein size, which tends to result in increased chemical shift overlap. Such overlap may result in ambiguous spin systems, where a TOCSY peak matches more than one spin system. Unfortunately, one cannot use fewer proton types while also using backbone nitrogen chemical shifts because it results in poor typing accuracy unless carbon chemical shifts are also used [29]. The test set obtained from the authors of the CR method did not have any ambiguous spin systems. For secondary structure type

prediction, obtaining a high yield of J-coupling constants from HNHA becomes more difficult as the protein size increases. An increase in size also tends to lead to an increase in the noise level of the edges, so correct contact matches may become lost. The level is increased further if automatically picked peaks are used. Using such peak lists for Ubiquitin yielded a noise level above 12 versus 4.6 from manually picked peaks. Rather than predicting a spin system's amino acid and secondary structure type, it might be simpler to exploit the known assignments in the BMRB, and predict a putative set of amino acids for each spin system. Promising preliminary results were obtained by using BMRB chemical shift data,  $H^\alpha$  from TOCSY, and protein structure information for building an interaction graph with a reduced noise level. Although this requires information about previous assignments, such information will become available during NMR studies involving different mutants of a given protein once one assignment has been determined.

## Acknowledgment

We would like to thank Xiong, Pandurangan, Bailey-Kellogg for providing us with their program and the test data for 5 proteins. We are grateful to Thorsten Dieckmann for thoughtful discussions. NMR spectra for TM1112 were generated as part of the US NIH Protein Structure initiative and kindly provided by A. Gutmanas and C. Arrowsmith.

This work is partially supported by NSERC Grant OGP0046506, China's MOST 863 Grant 2008AA02Z313, Canada Research Chair program, MITACS, an NSERC Collaborative Grant, Premier's Discovery Award, SHARCNET, and the Cheriton Scholarship.

## References

1. Alipanahi, B., Gao, X., Karakoc, E., Balbach, F., Donaldson, L., Arrowsmith, C., Li, M.: IPASS: Error tolerant NMR backbone resonance assignment by linear programming. Technical Report CS-2009-16, David R. Cheriton School of Computer Science, University of Waterloo, ON (2009), <http://www.cs.uwaterloo.ca/research/tr/2009/>
2. Alipanahi, B., Gao, X., Karakoc, E., Donaldson, L., Li, M.: PICKY: A novel SVD-based NMR spectra peak picking method. *Bioinformatics* 25, 268–275 (2009)
3. Altieri, A.S., Byrd, R.A.: Automation of NMR structure determination of proteins. *Curr. Opin. Struct. Biol.* 14(5), 547–553 (2004)
4. Apaydin, M.S., Conitzer, V., Donald, B.R.: Structure-based protein NMR assignments using native structural ensembles. *J. Biomol. NMR* 40(4), 263–276 (2008)
5. Bailey-Kellogg, C., Widge, A., Kelly, J., Brushweller, J., Donald, B.R.: The NOESY Jigsaw: Automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. *J. Comput. Biol.* 7, 537–558 (2000)
6. Bartels, C., Güntert, P., Billeter, M., Wüthrich, K.: GARANT - A general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. *J. Comput. Chem.* 18, 139–149 (1997)

7. Billeter, M., Wagner, G., Wüthrich, K.: Solution NMR structure determination of proteins revisited. *J. Biomol. NMR* 42(3), 155–158 (2008)
8. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: *Handbook of Combinatorial Optimization*, pp. 1–74. Kluwer Academic Publishers, Dordrecht (1999)
9. Burkard, R., Dell’Amico, M., Martello, S.: *Assignment Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2009)
10. Coggins, B.E., Zhou, P.: PACES: Protein sequential assignment by computer-assisted exhaustive search. *J. Biomol. NMR* 26(2), 93–111 (2003)
11. Danna, E., Felon, M., Gu, Z., Wunderling, R.: Generating multiple solutions for mixed integer programming problems. *Integer Programming and Combinatorial Optimization*, 280–294 (2007)
12. Drenth, J.: *Principles of Protein X-Ray Crystallography*, 3rd edn. Springer, Heidelberg (2007)
13. Erdmann, M.A., Rule, G.S.: Rapid protein structure detection and assignment using residual dipolar couplings. Technical Report CMU-CS-02-195, School of Computer Science, Carnegie Mellon University (2002)
14. Fiorito, F., Herrmann, T., Damberger, F.F., Wüthrich, K.: Automated amino acid side-chain NMR assignment of proteins using  $^{13}\text{C}$ - and  $^{15}\text{N}$ -resolved 3D [1H, 1H]-NOESY. *J. Biomol. NMR* 42(1), 23–33 (2008)
15. Goddard, T.D., Kneller, D.G.: *Sparky 3*. University of California, San Francisco
16. Greistorfer, P., Lokketangen, A., Vob, S., Woodruff, D.: Experiments concerning sequential versus simultaneous maximization of objective function and distance. *Journal of Heuristics* 14(6), 613–625 (2008)
17. Grishaev, A., Steren, C.A., Wu, B., Pineda-Lucena, A., Arrowsmith, C., Llinas, M.: Abacus, a direct method for protein NMR structure computation via assembly of fragments. *Proteins* 61(1), 36–43 (2005)
18. Gronwald, W., Willard, L., Jellard, T., Boyko, R.F., Rajarathnam, K., Wishart, D.S., Sönnichsen, F.D., Sykes, B.D.: CAMRA: Chemical shift based computer aided protein NMR assignments. *J. Biomol. NMR* 12(3), 395–405 (1998)
19. Güntert, P., Salzmann, M., Braun, D., Wüthrich, K.: Sequence-specific NMR assignment of proteins by global fragment mapping with the program MAPPER. *J. Biomol. NMR* 18(2), 129–137 (2000)
20. Harris, R.: The Ubiquitin NMR Resource Page, <http://www.biochem.ucl.ac.uk/bsm/nmr/ubq/index.html>
21. Hus, J., Prompers, J.J., Brüschweiler, R.: Assignment strategy for proteins with known structure. *J. Magn. Reson.* 157(1), 119–123 (2002)
22. Jung, Y., Zweckstetter, M.: Backbone assignment of proteins with known structure using residual dipolar couplings. *J. Biomol. NMR* 30(1), 25–35 (2004)
23. Jung, Y., Zweckstetter, M.: MARS – robust automatic backbone assignment of proteins. *J. Biomol. NMR* 30(1), 11–23 (2004)
24. Kamisetty, H., Bailey-Kellogg, C., Pandurangan, G.: An efficient randomized algorithm for contact-based NMR backbone resonance assignment. *Bioinformatics* 22(2), 172–180 (2006)
25. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)
26. Langmead, C.J., Donald, B.R.: An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *J. Biomol. NMR* 29(2), 111–138 (2004)



27. Langmead, C.J., Yan, A., Lilien, R., Wang, L., Donald, B.R.: A polynomial-time nuclear vector replacement algorithm for automated NMR resonance assignments. *J. Comput. Biol.* 11(2-3), 277–298 (2004)
28. Lemak, A., Steren, C.A., Arrowsmith, C.H.: Sequence specific resonance assignment via Multicanonical Monte Carlo search using an ABACUS approach. *J. Biomol. NMR* 41(1), 29–41 (2008)
29. Marin, A., Malliavin, T.E., Nicolas, P., Delsuc, M.-A.: From NMR chemical shifts to amino acid types: investigation of the predictive power carried by nuclei. *J. Biomol. NMR* 30(1), 47–60 (2004)
30. Meiler, J., Baker, D.: Rapid protein fold determination using unassigned NMR data. *Proc. Natl. Acad. Sci. U.S.A.* 100(26), 15404–15409 (2003)
31. Mittermaier, A., Kay, L.E.: New tools provide new insights in NMR studies of protein dynamics. *Science* 312(5771), 224–228 (2006)
32. Moseley, H.N., Sahota, G., Montelione, G.T.: Assignment validation software suite for the evaluation and presentation of protein resonance assignment data. *J. Biomol. NMR* 28(4), 341–355 (2004)
33. Moulton, J., Fidelis, K., Kryshchuk, A., Rost, B., Hubbard, T., Tramontano, A.: Critical assessment of methods of protein structure prediction (CASP): Round VII. *Proteins* 69, 3–9 (2007)
34. Moulton, J., Fidelis, K., Rost, B., Hubbard, T., Tramontano, A.: Critical assessment of methods of protein structure prediction (CASP): Round VI. *Proteins* 61, 3–7 (2005)
35. Pellecchia, M., Bertini, I., Cowburn, D., Dalvit, C., Giralt, E., Jahnke, W., James, T.L., Homans, S.W., Kessler, H., Luchinat, C., Meyer, B., Oschkinat, H., Peng, J., Schwalbe, H., Siegal, G.: Perspectives on NMR in drug discovery: a technique comes of age. *Nat. Rev. Drug Discov.* (August 2008)
36. Pintacuda, G., Keniry, M.A., Huber, T., Park, A.Y., Dixon, N.E., Otting, G.: Fast structure-based assignment of  $^{15}\text{N}$  HSQC spectra of selectively  $^{15}\text{N}$ -labeled paramagnetic proteins. *J. Am. Chem. Soc.* 126(9), 2963–2970 (2004)
37. Pons, J.L., Delsuc, M.A.: RESCUE: An artificial neural network tool for the NMR spectral assignment of proteins. *J. Biomol. NMR* 15(1), 15–26 (1999)
38. Pristovsek, P., Franzoni, L.: Stereospecific assignments of protein NMR resonances based on the tertiary structure and 2D/3D NOE data. *J. Comput. Chem.* 27(6), 791–797 (2006)
39. Pristovsek, P., Rüterjans, H., Jerala, R.: Semiautomatic sequence-specific assignment of proteins based on the tertiary structure - the program st2nmr. *J. Comput. Chem.* 23, 335–340 (2002)
40. Raymond, J.W., Willett, P.: Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J. Comput.-Aided Mol. Des.* 16(7), 521–533 (2002)
41. Powers, R., Mercier, K.A., Copeland, J.C.: The application of FAST-NMR for the identification of novel drug discovery targets. *Drug Discov. Today* 13(3-4), 172–179 (2008)
42. Skinner, A.L., Laurence, J.S.: High-field solution NMR spectroscopy as a tool for assessing protein interactions with small molecule ligands. *J. Pharm. Sci.* 97(11), 4670–4695 (2008)
43. Stratmann, D., Heijenoort, C., Guittet, E.: NOE-net—use of NOE networks for NMR resonance assignment of proteins with known 3D structure. *Bioinformatics* 25(4), 474–481 (2009)



44. Ulrich, E.L., Akutsu, H., Doreleijers, J.F., Harano, Y., Ioannidis, Y.E., Lin, J., Livny, M., Mading, S., Maziuk, D., Miller, Z., Nakatani, E., Schulte, C.F., Tolmie, D.E., Wenger, R.K., Yao, H., Markley, J.L.: BioMagResBank. *Nucleic Acids Res.* 36(Database issue), D402–D408 (2008)
45. Wang, A.C., Bax, A.: Determination of the backbone dihedral angles phi in human ubiquitin from reparametrized empirical Karplus equations. *J. Am. Chem. Soc.* 118(10), 2483–2494 (1996)
46. Wu, K., Chang, J., Chen, J., Chang, C., Wu, W., Huang, T., Sung, T., Hsu, W.: RIBRA—An error-tolerant algorithm for the NMR backbone assignment problem. *J. Comput. Biol.* 13(2), 229–244 (2006)
47. Wüthrich, K.: *NMR of Proteins and Nucleic Acids*. John Wiley & Sons, New York (1986)
48. Xia, Y., Yee, A., Semesi, A., Arrowsmith, C.H.: Solution structure of hypothetical protein TM1112. PDB Database (2002)
49. Xiong, F., Bailey-Kellogg, C.: A hierarchical grow-and-match algorithm for backbone resonance assignments given 3D structure. In: BIBE 2007, pp. 403–410 (2007)
50. Xiong, F., Pandurangan, G., Bailey-Kellogg, C.: Contact replacement for NMR resonance assignment. *Bioinformatics* 24(13), 205–213 (2008)
51. Zimmerman, D.E., Kulikowski, C.A., Huang, Y., Feng, W., Tashiro, M., Shimotakahara, S., Chien, C., Powers, R., Montelione, G.T.: Automated analysis of protein NMR assignments using methods from artificial intelligence. *J. Mol. Biol.* 269(4), 592–610 (1997)

# Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra

Kyowon Jeong<sup>1</sup>, Sangtae Kim<sup>2</sup>, Nuno Bandeira<sup>2</sup>, and Pavel A. Pevzner<sup>2</sup>

<sup>1</sup> Department of Electrical and Computer Engineering,  
University of California, San Diego, CA  
kwj@ucsd.edu

<sup>2</sup> Department of Computer Science and Engineering,  
University of California, San Diego, CA  
{sak008,bandeira,pevzner}@ucsd.edu

**Abstract.** Generating all plausible *de novo* interpretations of a peptide tandem mass (MS/MS) spectrum (Spectral Dictionary) and quickly matching them against the database represent a recently emerged alternative approach to peptide identification. However, the sizes of the Spectral Dictionaries quickly grow with the peptide length making their generation impractical for long peptides. We introduce Gapped Spectral Dictionaries (all plausible *de novo* interpretations with gaps) that can be easily generated for any peptide length thus addressing the shortcoming of the Spectral Dictionary approach. We show that Gapped Spectral Dictionaries are small thus opening a possibility of using them to speed-up MS/MS database searches. Our MS-GappedDictionary algorithm (based on Gapped Spectral Dictionaries) enables proteogenomics applications that are prohibitively time consuming with existing approaches. We further introduce gapped tags that have advantages over the conventional peptide sequence tags in filtration-based MS/MS database searches.

## 1 Introduction

Most peptide identification tools are still rather slow since they match every tandem mass (MS/MS) spectrum against all peptides in a database. A faster approach would be to generate a *de novo* reconstruction of a spectrum and to match the resulting peptide against a database. The fundamental algorithmic advantage of the latter approach is that one can pre-process the database (e.g., by constructing its suffix tree) so that matching becomes instantaneous. The only reason why most MS/MS database search tools still use the former approach is because *de novo* peptide sequencing remains inaccurate. Even the most advanced *de novo* peptide sequencing tools [6,7,16] correctly reconstruct only 30 - 45% of the *complete* peptides identified in MS/MS database searches. After decades of algorithmic developments, it seems that *de novo* peptide sequencing “hits a wall” and that accurate *full-length* peptide reconstruction is nearly impossible due to the limited *information content* of MS/MS spectra. We argue that regions with

low information content should be represented as mass gaps (the mass that may represent two or more amino acids) and advocate use of gapped peptides as spectral interpretations.

Kim et al., 2009 [13] recently proposed to generate *multiple de novo* reconstructions (rather than a single one) and to match them against a database (MS-Dictionary approach). Since matching peptides against a pre-processed database is very fast, generating thousands of reconstructions still has advantages over the traditional approaches where spectra are matched against large databases. Given an MS/MS spectrum, MS-Dictionary generates the *Spectral Dictionary* [13] that contains all plausible *de novo* reconstructions of the spectrum (i.e., with scores exceeding a given threshold) and further matches them against a database. The running time of MS-Dictionary is almost independent of the database size making it a tool of choice for peptide identification in large databases [13].

Although MS-Dictionary was proved to be useful for peptides shorter than 15 amino acids (aa), it has limitations for longer peptides with large Spectral Dictionaries. We introduce MS-GappedDictionary that generates rather small *Gapped Spectral Dictionaries* (even for long peptides) thus addressing the key limitation of the Spectral Dictionaries. Gapped Spectral Dictionary is the set of *gapped peptides* (see [14]) that are derived from the full-length peptides in the Spectral Dictionary. Gapped peptides occupy a niche between accurate but short peptide sequence tags [17] and long but inaccurate full-length peptide reconstructions. The gapped peptides are both long and accurate making them well suited for *de novo*-based MS/MS database search approaches. In difference from short peptide sequence tags, a gapped peptide typically has a single match in a database reducing peptide identification to a single database look-up. For a typical 20-aa long peptide, the size of Spectral Dictionary exceeds  $10^{17}$ , while the size of the Gapped Spectral Dictionary is only  $10^4$ . Moreover, we show that even smaller Gapped Spectral Dictionaries with only 20-100 peptides are sufficient for most applications. At the same time, gapped peptides are sufficiently long for efficient database matching. For example, for a spectrum of 15-aa long peptide, the average length<sup>1</sup> of gapped peptides in its Gapped Spectral Dictionary exceeds 9 aa. For all practical purposes, (gapped) peptides of length 9 are as informative as (full-length) peptides of length 15 for matching databases (unless the database size approaches  $20^9$ ). Table 1 (left upper panel) shows the Gapped Spectral Dictionary of a spectrum of peptide LNRVSQ GK consisting of 7 gapped peptides (as compared to its Spectral Dictionary consisting of 92 peptides shown in Table S1 in the Supplement). We describe an efficient algorithm for constructing the Gapped Spectral Dictionaries (using the *generating function* approach [12]) that also computes *coverage* of each gapped peptide, reflecting the portion of plausible *de novo* reconstructions represented by a gapped peptide (see section 3.2 for the definition of coverage).

Recent proteogenomics studies highlighted the importance of MS/MS searches against the six-frame translation of genomes [5,9,10,19]. However, until recently,

---

<sup>1</sup> The number of gaps and amino acids in the gapped peptide. For example, the length of [186]DK[246]FK is 6.

**Table 1. Left upper panel:** The Gapped Spectral Dictionary for the spectrum of peptide LNRVSQ GK (consisting of 7 gapped peptides) is much smaller than the Spectral Dictionary (consisting of 92 full-length peptides). For simplicity, LNRVSQ GK is represented by its *integer* amino acid masses as follows: [113][114][156][99][87][128][57][128]. Each gapped peptide is represented by amino acids and *mass gaps*. Note that a mass may represent combinations of amino acids (for example, [128] can be Q, K, GA, or AG). Either Q or K is used instead of [128] when [128] occupies the same position as Q or K on the peptide LNRVSQ GK. The gapped peptides that match the correct peptide are called *correct* gapped peptides. For example, the gapped peptides [113 + 114]RVSQ GK or LN[156+99]SQ GK match peptide LNRVSQ GK. In this Gapped Spectral Dictionary, the gapped peptides 1 and 6 (with <sup>†</sup>) are *correct* gapped peptides. The gapped peptides are shown in the descending order of their coverages, the portion of the total probability of all peptides in the Spectral Dictionary represented by a gapped peptide.

**Left lower panel:** Peptide sequence tags of length 3 derived from the Gapped Spectral Dictionary. Masses over left (right) arrows are the prefix (suffix) masses of the tags. Only 2 tags (e.g., QGK and VRV) cover all gapped peptides in the Gapped Spectral Dictionary.

**Right panel:** The Gapped Spectral Dictionary for the spectrum of peptide AIIDAIVS-GELK (16 gapped peptides represent 24,034 full length peptides). The correct gapped peptides are marked by <sup>†</sup>. The Gapped Spectral Dictionary for the peptide AIIDAIVS-GELK reveals only 3 tags (GEL, ELK, and SGE), together covering only 18.59% of the Spectral Dictionary.

No.	Gapped Peptide (GP)	Coverage of GP (%) <sup>*</sup>	# of peptides representing GP
1 <sup>†</sup>	[227]RVSQ GK	45.69	12
2	[128][255]VSQ GK	15.99	32
3	[128]VRVSQ GK	13.71	20
4	[128]VR[186]Q GK	11.42	4
5	[128]VRV[215]G K	5.71	2
6 <sup>†</sup>	[383]VSQ GK	5.71	2
7	[128]G[198]VSQ GK	1.77	20
Total	-	100	92

No.	Tag	Tag coverage(%)	Covered GP
1	569 QGK 0	94.3	1,2,3,4,6,7
2	383 VSQ 185	82.9	1,2,3,6,7
3	482 SQG 128	82.9	1,2,3,6,7
4	227 RVS 313	59.4	1, 3
5	128 VRV 400	19.4	3, 5

No.	Gapped Peptide (GP)	Coverage of GP (%) <sup>*</sup>	# of peptides representing GP
1	[445][250]S[186]LK	33.81	3286
2 <sup>†</sup>	[695]S[186]LK	19.18	1703
3	[445][337][186]LK	13.28	255
4	[445][250][273]LK	7.67	178
5 <sup>†</sup>	[782]GELK	6.10	684
6 <sup>†</sup>	[695]SGELK	5.55	5563
7	[445][250]S[299]K	4.20	901
8	[445][250]SGELK	3.78	3437
9	[445][337]GELK	1.98	1072
10	[445][250]SG[242]K	1.61	3942
11 <sup>†</sup>	[695]SG[242]K	0.91	1614
12	[445][394]ELK	0.91	507
13	[445][250]SG[370]	0.66	604
14	[445][250][144]ELK	0.20	91
15 <sup>†</sup>	[695][144]ELK	0.07	35
16	[445][337]G[242]K	0.09	162
Total	-	100	24034

searches against the six-frame translations of large genomes were impractical even with the fastest MS/MS search tools. Although MS-Dictionary enabled searches in the six-frame translation of the human genome with 40X speed-up over InsPecT [13], it loses many peptide identifications because Spectral Dictionaries of long peptides have to be truncated (leading to truncating the correct peptides in some cases). Gapped Spectral Dictionaries remedy this shortcoming of Spectral Dictionaries and nearly double the number of identified peptide in the six-frame translation of the human genome (as compared to MS-Dictionary).

Table 1 (left lower panel) illustrates how gapped peptides and their coverage can be utilized for constructing the *peptide sequence tags* [17]. Tanner et al., 2005 [20] introduced *covering sets* of tags (set of tags containing at least

one correct tag) and demonstrated how such sets can greatly speed-up MS/MS database searches. However, while the sizes of covering sets may vary between spectra, Tanner et al., 2005 [20] did not describe an approach for selecting (the varying number of) tags for every spectrum and did not assign rigorous probabilities to tags. As a result, InsPecT currently generates a fixed number of tags for each spectrum. While Gapped Spectral Dictionaries can be utilized for generating (varying number of) conventional peptide sequence tags along with their probabilities, Table 1 (right panel) illustrates that “good” peptide sequence tags (representing all peptides in the Gapped Spectral Dictionary) may be difficult to find. We advocate generating *gapped* tags representing sequences of mass gaps (like [186]LK derived from the first peptide in the right panel of Table 1) and demonstrate that gapped tags improve the filtration efficiency of peptide sequence tags in tag-based MS/MS database searches.

## 2 Methods

### 2.1 Path Dictionary Problem

Most *de novo* peptide sequencing algorithms interpret spectra by analyzing paths in *spectrum graphs* [2]. We start by discussing the problem of finding suboptimal paths in *arbitrary* graphs and later describe how it relates to finding paths in the spectrum graphs.

Let  $G(V, E, score, probability)$  be a *directed acyclic graph* with vertex set  $V$ , edge set  $E$ , and functions *score* and *probability* defined on its edges. Given a path in  $G$ , the *score* of the path is defined as the sum of scores of its edges, while the *probability* of the path is defined as the product of probabilities of its edges. Given a graph  $G$  with selected vertices  $s$  (*source*) and  $t$  (*sink*), and a threshold  $MinScore$ , the *Path Dictionary* (denoted as  $PD(G, MinScore)$ ) is defined as the set of all paths from  $s$  to  $t$  with scores exceeding  $MinScore$  (along with their probabilities). The following Path Dictionary Problem can be solved using standard algorithms for finding suboptimal paths [4].

**Path Dictionary Problem.** Given a directed acyclic graph  $G$  and a threshold  $MinScore$ , construct  $PD(G, MinScore)$ .

Define  $p(x)$  as the total probability of all paths of score  $x$  from the source  $s$  to the sink  $t$  in the graph  $G$ . The *generating function*  $x \rightarrow p(x)$  can be efficiently computed as the probability of node  $(t, x)$  in the *dynamic programming graph* as described in [12, 13] (Figure 1, left).  $PD(G, MinLength)$  is constructed by standard *backtracking* in the dynamic programming graph.

For a *spectrum graph* of a tandem (MS/MS) mass spectrum [2], the Path Dictionary Problem corresponds to *de novo* peptide sequencing problem when multiple (suboptimal) *de novo* reconstructions (rather than a single one) are generated. Kim et al., 2008 [12] applied the generating function approach (Figure 1, left) to analyze MS/MS spectra and further demonstrated [13] how to generate the Path Dictionary (termed *Spectral Dictionary*) that contains *all* plausible *de novo* reconstructions for a given spectrum. Each path in Path Dictionary corresponds to a full-length peptide reconstruction in the Spectral Dictionary, and

$\sum_{x > \text{MinScore}} p(x)$  corresponds to *spectral probability*. To generate the Spectral Dictionaries, a spectral probability *Threshold* is fixed and *MinScore* is selected in such a way that the spectral probability does not exceed *Threshold*.

This Spectral Dictionary approach, while useful, is not practical for long peptides (15 amino acids and longer) with large dictionaries. We bypass this problem by solving the *Gapped Path Dictionary Problem* defined below.

## 2.2 Gapped Path Dictionary Problem

Let  $H$  be a subset of vertices of a graph  $G$  containing the source  $s$  and the sink  $t$  (vertices of  $H$  are called *hubs*). We remark that every path on vertices in  $G$  induces a *hub path* on vertices in  $H$  by simply retaining only vertices from  $H$  in the original path. For example, a path  $s \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow t$  that contains hubs  $s, v_2, v_3, v_5, t$  induces a hub path  $s \rightarrow v_2 \rightarrow v_3 \rightarrow v_5 \rightarrow t$ . We define the probability of a hub path as the total probability of all paths inducing this hub path. The Gapped Path Dictionary  $GPD(G, H, \text{MinScore})$  is defined as the set of all hub paths induced by the paths in  $PD(G, \text{MinScore})$  (along with their probabilities).

**Gapped Path Dictionary Problem.** Given a directed acyclic graph  $G$ , a subset of its vertices  $H$ , and a threshold *MinScore*, construct the Gapped Path Dictionary  $GPD(G, H, \text{MinScore})$ .

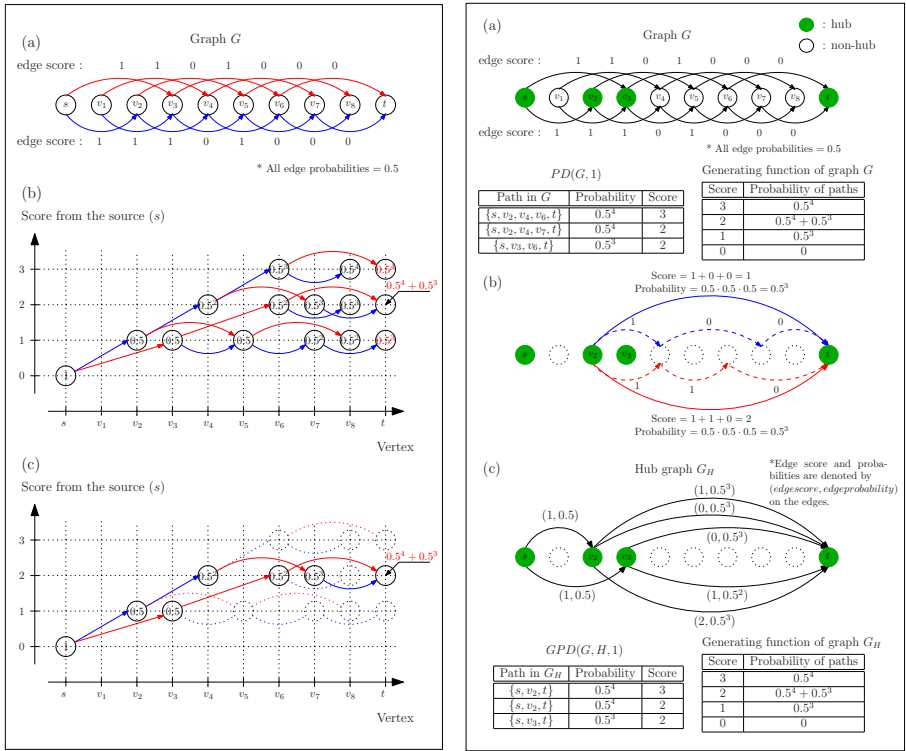
The brute-force algorithm for constructing  $GPD(G, H, \text{MinScore})$  (by constructing  $PD(G, \text{MinScore})$  and generating all hub paths induced by the paths in  $PD(G, \text{MinScore})$ ) is impractical for large  $PD(G, \text{MinScore})$ . Below we describe an efficient algorithm for solving the Gapped Path Dictionary Problem that does not require the construction of  $PD(G, \text{MinScore})$ .

Given hubs  $h$  and  $h'$ , we define  $\text{Path}(h, h')$  as the set of all paths in  $G$  between  $h$  and  $h'$  that *do not pass* through other hubs. Each path in  $\text{Path}(h, h')$  is characterized by its score and probability. Let  $\mathcal{X}(h, h')$  be the set of scores of all paths from  $\text{Path}(h, h')$  and  $\text{Prob}(h, h')$  be the total probability of all paths in  $\text{Path}(h, h')$ . If  $\text{Prob}(h, h', x)$  is defined as the total probability of all paths of score  $x$  from the set  $\text{Path}(h, h')$ , then  $\text{Prob}(h, h') = \sum_{x \in \mathcal{X}(h, h')} \text{Prob}(h, h', x)$ .

We define the *hub graph*  $G_H$  as a multigraph on the set of vertices  $H$  (Figure 1, right panel). For every  $x \in \mathcal{X}(h, h')$ , there exists an edge between  $h$  and  $h'$  with score  $x$  and probability  $\text{Prob}(h, h', x)$ <sup>2</sup>. The *score* and the *probability* of a path in  $G_H$  is defined as the sum of scores and the product of probabilities of its edges, respectively.

As the hub paths (on vertices in  $H$ ) are induced by the paths in  $G$ ,  $GPD(G, H, \text{MinScore})$  is the same as  $PD(G_H, \text{MinScore})$ . Therefore, the Gapped Path Dictionary Problem in  $G$  is essentially the *Path Dictionary Problem* in the hub graph  $G_H$ , and we only need to compute the scores and the probabilities of the edges in  $G_H$  to solve the Gapped Path Dictionary Problem. Below, we show how to compute  $\text{Prob}(h, h', x)$  for all edges of the hub graph.

<sup>2</sup> There exists  $|\mathcal{X}(h, h')|$  edges between vertices  $h$  and  $h'$  in the multigraph  $G_H$ .



**Fig. 1. Left panel:** Illustration of the dynamic programming algorithm for computing the generating function of graph  $G$  shown in (a). The nodes of the *dynamic programming* (DP) graph (b) are defined as pairs  $(v, x)$ , where  $v$  is a vertex of  $G$  and  $x$  is a score. Two nodes  $(v, x)$  and  $(v', x')$  are connected by an edge iff there exists an edge between vertices  $v$  and  $v'$  in  $G$  with score  $x' - x$ . The probability of an edge between  $(v, x)$  and  $(v', x')$  in the DP graph equals to the probability of the edge  $(v, v')$  in  $G$ . A source  $s$  in graph  $G$  corresponds to a single node  $(s, 0)$  in the DP graph. A node  $(v, x)$  is present in the DP graph iff there exist a path from  $(s, 0)$  to  $(v, x)$ . In this example, red (blue) edges of the DP graph in (b) are from the red (blue) edges of the graph  $G$  in (a). All edge probabilities in (b) are 0.5 as the probabilities of edges of  $G$  are 0.5. The *node probability* of node  $(v, x)$  (shown inside nodes in (b) and (c)) is the total probability of the paths from the source  $s$  to  $v$  with the score  $x$ . The node probability of the source of the DP graph is initialized by 1, and the node probability of a node  $(v, x)$  is obtained by the *weighted* summation of the node probabilities of its *predecessors* (see [12]). The generating function is represented by the probabilities of the sink nodes in the DP graph. To find all paths of score  $x$  from the source to the sink in graph  $G$  one has to backtrack all paths from the node  $(t, x)$  in the DP graph. If  $x = 2$ , two paths of score 2 are found as in (c):  $\{s, v_2, v_4, v_7, t\}$  and  $\{s, v_3, v_6, t\}$ . **Right panel:** Path Dictionary and Gapped Path Dictionary. (a)  $PD(G, 1)$  and the generating function of  $G$ . (b) The construction of  $G_H$  using edges between hubs  $v_2$  and  $t$  (shown as solid blue and red edges) as examples. Solid blue and red edges in  $G_H$  are induced by dashed blue and red paths in  $G$ . All paths that use only non-hub vertices in  $G$  are collapsed into edges in  $G_H$ . (c) The hub graph  $G_H$ ,  $GPD(G, H, 1)$ , and the generating function of  $G_H$ .



Given a hub  $h$  in the graph  $G(V, E, score, probability)$ , we modify the score function by assigning score  $-\infty$  to all edges originating at all hubs other than  $h$ . Denote the resulting score function (parameterized by  $h$ ) as  $score(h)$ . The family of score functions  $score(h)$  for all hubs  $h \in H$  can be used to compute  $Prob(h, h', x)$  for all pairs of hub vertices  $h$  and  $h'$ . One can prove that computing  $Prob(h, h', x)$  (for all  $x \in \mathcal{X}(h, h')$ ) is equivalent to computing the generating function for a graph  $G(V, E, score(h), probability)$  with source  $h$  and sink  $h'$ . Note that a single computation of the generating function from  $h$  to the sink  $t$  for the graph  $G(V, E, score(h), probability)$  gives us  $Prob(h, h', x)$  for all  $h' \in H$  and all  $x \in \mathcal{X}(h, h')$ .

After constructing the hub graph  $G_H$ ,  $GPD(G, H, MinScore)$  can be constructed by computing *generating function* for the graph  $G_H$  and generating all paths with score exceeding  $MinScore$ . Figure 1 shows an example of the Path Dictionary and the Gapped Path Dictionary.

### 2.3 Compact Gapped Path Dictionaries

So far, we represented each path in the Gapped Path Dictionary as the sequence of *edges* (rather than *vertices*) the path traverses. Since the hub graph  $G_H$  is a multigraph (that may have multiple edges of various scores between the same vertices), there can be many paths (with different scores) with identical *vertex-sets* (Figure 1, right panel (c)). We define the *Compact Gapped Path Dictionary*, denoted by  $CGPD(G, H, MinScore)$ , as the set of *vertex-sets* of paths in the Gapped Path Dictionary  $GPD(G, H, MinScore)$ , along with their *probabilities*, where the *probability* of each vertex-set in  $CGPD(G, H, MinScore)$  is defined as the total probability of the paths in  $GPD(G, H, MinScore)$  with the same vertex-set (see Table S1 in the Supplement).

The Compact Gapped Path Dictionary  $CGPD(G, H, MinScore)$  can be generated (albeit inefficiently) from  $GPD(G, H, MinScore)$  by simply representing all paths with identical vertex-sets as a single vertex-set and adding up the probabilities of all such paths. However, one can efficiently generate the Compact Gapped Path Dictionary without explicitly constructing Gapped Path Dictionary. Since the Gapped Path Dictionary in  $G$  is the same as the Path Dictionary in the multi-graph  $G_H$ , the Compact Gapped Path Dictionary in  $G$  is the same as the vertex-sets of Path Dictionary in  $G_H$ . It is easy to see that generating these vertex-sets can be achieved by retaining only the edges with the highest scores among parallel edges in the (multi)graph  $G_H$  and constructing the Path Dictionary in the resulting (simple) graph. The Path Dictionary in this modified  $G_H$  induces the vertex-sets of the Compact Gapped Path Dictionary in  $G$ .

After the Compact Gapped Path Dictionary is generated, one still needs to compute the probability of each vertex-set. This again can be done by applying MS-GeneratingFunction to a graph consisting of a single path corresponding to each vertex-set in the Compact Gapped Path Dictionary (this path represents a multi-graph since it may contain parallel edges). The probability of the gapped peptide represented by a vertex-set is given by the summation of the probabilities of all edge-paths (with the same vertex-set) with scores exceeding  $MinScore$ .



## 2.4 Gapped Spectral Dictionaries

For each spectrum, we construct its spectrum graph and generate a set of hubs (prefix masses). Given a spectrum graph  $G$  and a set of hubs  $H$ , paths in  $G$  correspond to peptides while vertex-sets in  $G_H$  correspond to *gapped peptides* introduced in [14]. *Gapped Spectral Dictionary* is defined as Compact Gapped Path Dictionary of the spectrum graph.

While we described an algorithm for constructing the Gapped Spectral Dictionary for a given hub set  $H$ , it remains unclear how to select hubs. The hub selection has to achieve two conflicting goals: (i) minimize the number of selected hubs to ensure that the constructed Gapped Spectral Dictionary is small, and (ii) maximize the average length of peptides in the Compact Gapped Spectral Dictionary to ensure that the reconstructed gapped peptides are sufficiently informative.

Therefore, the goal is to select  $k$  hubs that maximize the average number of vertices per path in the Gapped Path Dictionary (weighted by their probabilities). We select hubs as  $k$  most “popular” vertices in paths in  $PD(G, MinScore)$ . Such ranking of vertices of the graph  $G$  can be computed by generating *Spectral Profiles* introduced in [14] [3].

## 3 Results

### 3.1 Datasets

We used the *previously analyzed* Shewanella [12], HEK [13] and Standard [15] datasets to benchmark MS-GappedDictionary. **Shewanella** dataset (18,468 doubly charged spectra of distinct tryptic peptides from *Shewanella oneidensis* MR-1) is used to benchmark the performance of MS-GappedDictionary. **HEK** dataset (21,605 charge 2 spectra of distinct human peptides) is used to test database searches with MS-GappedDictionary. **Standard** dataset (990 charge 2 spectra of distinct human peptides) is used to benchmark applications of MS-GappedDictionary for (gapped) tag generation. See Supplement for the detailed description of these datasets.

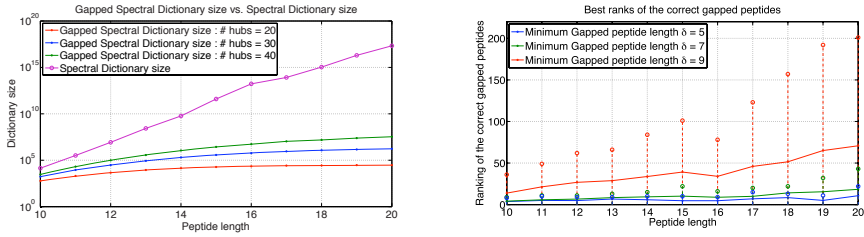
To generate the Gapped Spectral Dictionaries, the value of the spectral probability threshold is set to  $10^{-9}$  for Shewanella and Standard datasets and  $10^{-11}$  for HEK dataset (assuming that the precursor integer mass is known). The spectral hubs are selected based on  $k$  maximal peaks in its Spectral Profile with  $k$  varying from 20 to 40.

### 3.2 From Gapped Spectral Dictionaries to Pocket Dictionaries

Since multiple peptides often induce the same gapped peptide, Gapped Spectral Dictionaries are typically much smaller than Spectral Dictionaries. Figure 2

---

<sup>3</sup> The Spectral Profiles provide a better hub selection than peak intensities and PRMs [20] (see Supplement).



**Fig. 2. Left panel:** Gapped Spectral Dictionary size vs. Spectral Dictionary size (for varying peptide length and number of hubs) in Shewanella dataset.

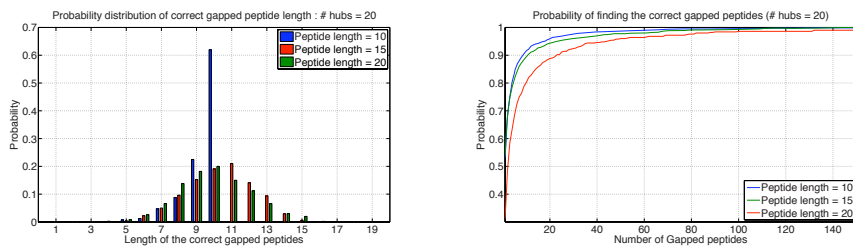
**Right panel:** Coverage ranking of (best ranked) correct gapped peptides in Shewanella dataset. We have chosen the best ranked correct gapped peptide for each spectrum (There can be more than one correct gapped peptides per each spectrum as in Table [1](#)). The average ranking does not exceed 80 regardless of the peptide length (for  $\delta = 5, 7, 9$ ). The number of hubs is 20. The bubbled stems represent the range that the rankings fall into with probability 90%.

(left) shows the sizes of Gapped Spectral Dictionaries and Spectral Dictionaries for various peptide lengths. While the size of Spectral Dictionary grows as  $20^{\text{peptide length}}$ , the size of the Gapped Spectral Dictionary is limited by  $2^{|H|}$ , where  $|H|$  is the number of hubs. In practice, the size of Gapped Spectral Dictionaries is much smaller than  $2^{|H|}$  for sensible values of spectral probabilities. For example, for peptides of length 20, the size of the Spectral Dictionary exceeds  $10^{17}$  while the size of the Gapped Spectral Dictionary is on the order of  $10^4$  (for  $|H| = 20$ ).

Figure [3](#) (left) shows the probability distribution of the lengths of the gapped peptides that are induced by the correct peptides (*correct gapped peptides*). The probability that these gapped peptides are short (length less than 5) is less than 0.01 regardless of the peptide length. The high average length of the correct gapped peptides (10 - 13) indicates that Gapped Spectral Dictionaries have the potential to speed up database searches. Gapped peptides are classified into *short* (with length shorter than  $\delta$ ) and *long* (with length equal to or longer than  $\delta$ ). Discarding short gapped peptides results in  $\delta$ -reduced Gapped Spectral Dictionary (with minimum gapped peptide length  $\delta$ ).

A spectrum is  $\delta$ -identifiable if its  $\delta$ -reduced Gapped Spectral Dictionary contains at least one correct gapped peptide. Figure [S2](#) in the Supplement shows the identifiability of spectra in the Shewanella dataset. For 20 hubs and  $\delta = 5$ , the identifiability is higher than 99% for all peptide lengths. Figure [S2](#) illustrates that there exists a tradeoff between the identifiability and efficiency of the database search controlled by the minimum length of the gapped peptide  $\delta$  (increase in  $\delta$  reduces the identifiability but improves the filtering efficiency of the database search).

After generating the  $\delta$ -reduced Gapped Spectral Dictionaries, we order all gapped peptides by their *coverages*, and analyze the rank of the first correct gapped peptides in this ranked list. The *coverage* of a gapped peptide is defined



**Fig. 3. Left panel:** Probability distribution of the length of the gapped peptide induced by correct peptides in Shewanella dataset.

**Right panel:** The probability that a correct gapped peptide is found within  $k$  top-ranked peptides in the  $\delta$ -reduced Gapped Spectral Dictionary. The number of hubs is 20, and  $\delta = 5$  (see Supplement for different parameters).

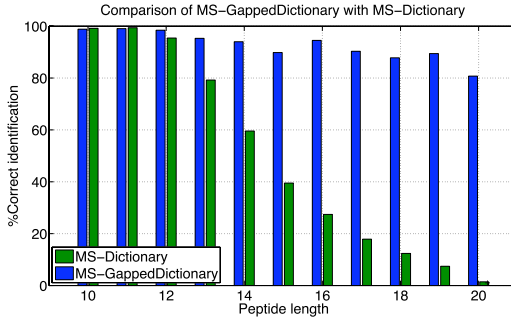
as the probability of the gapped peptide divided by the total probability of the peptides in the Spectral Dictionary. Figure 2, right shows that the average rank of the best ranked correct gapped peptides does not exceed 100 even for long gapped peptides ( $\delta = 5, 7, 9$ ). In fact, only 20 - 100 gapped peptides are typically sufficient to generate a *covering set* containing a correct peptide (Figure 3, right). As such, it suffices to generate a small subset of the Gapped Spectral Dictionary called *Pocket Dictionary* by choosing the  $k$  best-ranked gapped peptides in the  $\delta$ -reduced Gapped Spectral Dictionary ( $k$  is typically 20 - 100).

Figure S2 (right panel) in the Supplement shows the identifiability of the Pocket Dictionaries compared to the identifiability in the (full-size)  $\delta$ -reduced Gapped Spectral Dictionaries. Throughout the paper we generate Pocket Dictionaries of size 100 with  $\delta = 5$  and 20 hubs that result in high identifiability<sup>4</sup>

### 3.3 Database Search with Gapped Spectral Dictionaries

Figure 4 shows the percentage of spectra in the HEK dataset identified with MS-Dictionary and MS-GappedDictionary in searches against the six-frame translation of the human genome. For peptides of length 20, MS-GappedDictionary identified  $\approx 80\%$  of all peptides while MS-Dictionary identified only  $\approx 4\%$  of them. MS-GappedDictionary reliably identified 19,280 of 21,605 HEK spectra

<sup>4</sup> While we showed how to generate the *highest-scoring* gapped peptides, it is not immediately clear how to generate the *highest-probability* vertex-sets (gapped peptides) in the  $\delta$ -reduced Gapped Path Dictionary. This difficulty stems from the fact that the  $y$ -axis in the DP graph (Figure 1) represents accumulated *scores* and not accumulated *probabilities*. To address this problem, we implemented a depth-first branch-and-bound backtracking traversal of the DP graph that uses accumulated scores to determine membership in the Pocket Dictionary and accumulated probabilities to select the highest-coverage peptides. The algorithm maintains the accumulated probability for every suffix extension and combines it with node probabilities (Figure 1) to prune extensions whose maximum probability is lower than that of the current  $k$ -th ranked highest-probability peptide.



**Fig. 4.** The percentage of peptides in HEK dataset identified by MS-GappedDictionary and MS-Dictionary in the six-frame translation of the human genome as compared with peptides identified in searches of human protein database

in the six-frame translation of the human genome nearly doubling the number of peptides identified by MS-Dictionary (10,266 peptides were reported in [13]). It illustrates that MS-GappedDictionary finds  $\approx 90\%$  of peptides identified in human protein database by searching the 80-times large six-frame translation of the human genome (i.e., without knowing where the genes are). Therefore, MS-GappedDictionary significantly improves on MS-Dictionary in proteogenomics gene discovery and annotation.

In contrast to MS-Dictionary, peptides identified by MS-GappedDictionary may not belong to the Spectral Dictionary. For example, a gapped peptide AT[144]GG may match ATSGGG (in the Spectral Dictionary) and ATGSGG (not in the Spectral Dictionary). Thus, peptides matched by MS-GappedDictionary have to be scored to remove those that are not in the Spectral Dictionary. Since the number of matches reported by MS-GappedDictionary is typically small (Table S5), the time required for removing low-scoring peptides is negligible (less than 0.01 s per spectrum).

The current version of MS-GappedDictionary uses gapped tags to speed-up searches in huge databases. Below we sketch a more efficient algorithm (based on matching the entire gapped peptides) that will be described in detail elsewhere.

A brute-force algorithm to match the set of  $m$  gapped peptides against a database of size  $n$  has  $O(kmn)$  complexity, where  $k$  is the maximum length of peptides. The complexity can be further reduced by constructing the *keyword tree* of gapped peptides. We construct a *Master Dictionary* that combines the Pocket Dictionaries for all spectra. Since the Pocket Dictionaries have only 20 - 100 gapped peptides per spectrum, the size of the Master Dictionary is only 1-2 orders of magnitude larger than the size of the spectral dataset. We further construct the keyword tree of the Master Dictionary (in the alphabet of gap masses) in  $O(km)$  time (we assume constant alphabet size). For every peptide of length  $k$  in the database we generate all gapped peptides induced by this peptide (the number of such peptides is bounded by  $2^{k-1}$ ) and combine all such peptides into a Master Gapped Database. Since matching each gapped peptide

against the keyword tree of the Master Gapped Spectral Dictionary takes  $O(k)$  time, matching the entire Master Gapped Database against the keyword tree takes  $O(km + f(k)n)$  where  $f(k)$  is bounded by  $k \cdot 2^{k-1}$  ( $f(k)$  is expected to be much smaller in practice)<sup>5</sup>

### 3.4 From Gapped Spectral Dictionaries to Gapped Tags

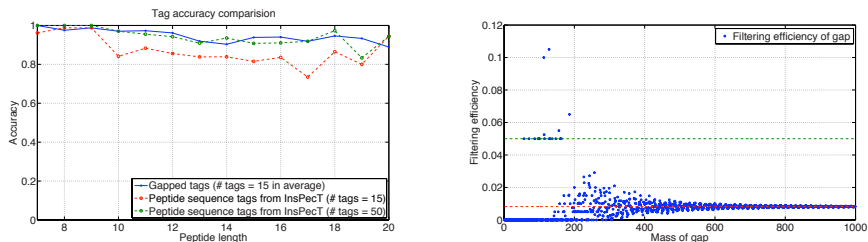
The right panel of Table 1 demonstrates that many gapped peptides in the Gapped Spectral Dictionary may not contain peptide sequence tags. In contrast, allowing a single gap in tags (*gapped tags*) reveals a covering set of only 6 tags of length 3 : [273]LK, G[242]K, S[299]K, [250]SG, ELK, and [186]LK. In contrast with peptide sequence tags, gapped tags include both gaps and amino acid masses. Below we limit our analysis to gapped tags with gaps below 500 Da<sup>6</sup> and analyze gapped tags of length 3 with at most one gap (i.e., gapped tags with at least 2 amino acids). Such tags are called *proper* gapped tags. We demonstrate that the proper gapped tags have better filtering efficiency than peptide sequence tags. Some masses in a gapped peptide may represent either an amino acid or a gap because 5 amino acids (N, Q, K, R, and W with masses 114, 128, 128, 156, and 186, respectively) have *composite* masses equal to the sum of two amino acid masses. For example, the composite mass 114 Da could represent either N or GG. One can check whether a composite mass represents an amino acid by examining the hub set. A mass  $m$  is a *submass* of a composite mass  $mass$  if both  $m$  and  $mass - m$  represent masses of amino acid. If  $mass$  starts at position  $prefixMass$  in a gapped peptide, then it represents an amino acid iff  $prefixMass + m$  represents a hub for each submass  $m$  of mass  $mass$ .

To generate the set of proper gapped tags, we select at most one proper gapped tag from each gapped peptide in the Pocket Dictionary. The greedy algorithm for selecting proper gapped tags is described in the Supplement. Figure 5 (left panel) compares the gapped tags generated by MS-GappedDictionary with peptide sequence tags generated by InsPecT. Using only 15 proper gapped tags generated by MS-GappedDictionary (see Table S4), the overall accuracy is 95.1% while the accuracy of InsPecT tags is only 87.2% with 15 peptide sequence tags and 94.7% even with 50 tags.

MS-GappedDictionary constructs a hash table using proper gapped tags from a database as the keys and their database positions as the hashed values (if a proper gapped *tag* starts at a database *position*, the value *position* is added to the key *tag*). This approach has a memory overhead since the hash table for gapped tags is larger than the hash table for the conventional peptide sequence tags. However, by limiting the gap size in the proper gapped tags to 500 Da, the memory increase can be tolerated. For example, the hash table of the Swiss-Prot

<sup>5</sup> Matching gapped peptides against a database can be formulated as the *pattern matching problem with don't cares* [18,18]. Implementing these algorithms will result in further speed-up of MS-GappedDictionary.

<sup>6</sup> We limit the mass of the largest gap to limit the memory requirements of MS-GappedDictionary (see below).



**Fig. 5. Left panel:** Comparison of gapped tags generated from the Pocket Dictionaries and the peptide sequence tags generated by InsPecT (in Standard dataset).

**Right panel:** Filtering efficiency of a mass gap. Each blue dot  $(x, y)$  denotes the filtering efficiency ( $y$ ) of the mass gap  $x$ . Typically, the filtering efficiency of a gap is smaller than that of an amino acid (shown as green dashed line  $y = \frac{1}{20}$ ), and as mass grows, it converges to  $\frac{1}{\text{average amino acid mass}}$  (shown as the red dashed line  $y = \frac{1}{121.6}$ ).

database (release 56.6, 146 million residues) is only 10 times larger than the size of the database.

Once the hash table is built, finding peptides matched to a proper gapped tag is fast, and the search space for further analysis is limited to only those matched peptides. We define the *filtration efficiency* of a gapped tag/peptide sequence tag/peptide as the ratio of the number of its matches in the random<sup>7</sup> database over the database size. While the filtration efficiency of a peptide is  $1/20^{\text{peptide length}}$  (and the filtration efficiency of an amino acid is  $1/20$ ), it is easy to see that the filtration efficiency of a gap of mass  $m$  is the sum of filtration efficiencies of all peptides with mass  $m$ . It turns out that (large) masses typically have better filtration efficiencies than amino acids. Figure 5 illustrates that the filtration efficiency of masses larger than 250 Da fluctuates around  $1/(\text{average amino acid mass})$  resulting in  $\approx 6$ -fold improvement in filtration efficiency as compared to amino acids. This improvement translates into a superior filtration efficiency of gapped tags as compared to peptide sequence tags.

For each spectrum of Standard dataset, we generated tags using MS-Gapped-Dictionary (15 proper gapped tags per spectrum) and InsPecT (50 peptide sequence tags per spectrum), and measured the number of tag matches against the Swiss-Prot database. While InsPecT reported  $\approx 2$  million peptide sequence tag matches, MS-GappedDictionary reported only  $\approx 450$  thousands gapped tag matches. It directly leads to the speed up of the database search. The running time to search the Swiss-Prot database was 0.21 sec for MS-GappedDictionary (including the generation of the Gapped Spectral Dictionary and the gapped tags) and 0.51 sec for InsPecT per spectrum on a desktop machine with a 2.67-GHz Intel processor. We also searched the six frame translation of the human genome and MS-GappedDictionary (0.36 sec per spectrum) showed  $\approx 20X$

<sup>7</sup> A database with identically and independently distributed amino acids with probability  $1/20$ .

speed-up as compared to InsPecT (8.5 sec per spectrum)<sup>8</sup> Note that InsPecT is one of the fastest database search tools that is 10 times faster than X!Tandem and 60 times faster than SEQUEST [13].

## 4 Discussion

Gapped peptides occupy a niche between accurate but short peptide sequence tags and long but inaccurate full-length peptide reconstructions. The gapped peptides are both long and accurate making them an ideal choice for *de novo*-based MS/MS database search approaches. In difference from peptide sequence tags, they typically have a single match in a database reducing peptide identification to a single look-up in the database. While future work will focus on efficient matching of full-length gapped peptides against large databases, we show how gapped tags can be generated from gapped peptides to effectively filter indexed databases. Furthermore, we show how the concept of *coverage* can be instrumental for ranking sparse representations of spectral dictionaries, here limited to gapped tags and gapped peptides but conceptually generalizable to any sparse representation of all plausible peptide reconstructions.

## References

1. Cole, R., Gottlieb, L.-A., Lewenstein, M.: Dictionary matching and indexing with errors and don't cares. In: STOC, pp. 91–100 (2004)
2. Dancik, V., Addona, T., Clauser, K., Vath, J., Pevzner, P.: De novo protein sequencing via tandem mass-spectrometry. *J. Comp. Biol.* 6, 327–341 (1999)
3. Eng, J., McCormack, A., Yates, J.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* 7, 655–667 (1994)
4. Eppstein, D.: Finding the k Shortest Paths. *SIAM J. Comput.* 28, 652–673 (1998)
5. Fermin, D., Allen, B., Blackwell, T., Menon, R., Adamski, M., Xu, Y., Ulintz, P., Omenn, G., States, D.: Novel gene and gene model detection using a whole genome open reading frame analysis in proteomics. *Genome Biol.* 7, R35 (2006)
6. Frank, A., Pevzner, P.: PepNovo: de novo peptide sequencing via probabilistic network modeling. *Anal. Chem.* 77, 964–973 (2005)
7. Frank, A.: A ranking-based Scoring Function for peptide-spectrum matches. *J. Proteome Res.* 8, 2241–2252 (2009)
8. Iliopoulos, C.S., Rahman, M.S.: Pattern Matching Algorithms with Don't Cares. In: SOFSEM 2007, pp. 116–126 (2007)
9. Jaffe, J., Berg, H., Church, G.: Proteogenomic mapping as a complementary method to perform genome annotation. *Proteomics* 4, 59–77 (2004)
10. Kalume, D., Peri, S., Reddy, R., Zhong, J., Okulate, M., Kumar, N., Pandey, A.: Genome annotation of *Anopheles gambiae* using mass spectrometry-derived data. *BMC Genomics* 6, 128–138 (2005)

---

<sup>8</sup> The hash table is computed once, stored as a file, and loaded before performing the search. The loading time becomes negligible (compared to the search time) for large data sets.

11. Keller, A., Nesvizhskii, A., Kolker, E., Aebersold, R.: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* 74, 5383–5392 (2002)
12. Kim, S., Gupta, N., Pevzner, P.: Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.* 7, 3354–3363 (2008)
13. Kim, S., Gupta, N.: Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. *Mol. Cell Proteomics* 8, 53–69 (2009)
14. Kim, S., Bandeira, N., Pevzner, P.: Spectral Profiles, a Novel Representation of Tandem Mass Spectra and Their Applications for de Novo Peptide Sequencing and Identification. *Mol. Cell Proteomics* 8, 1391–1400 (2009)
15. Klimek, J., Eddes, J.S., Hohmann, L., Jackson, J., Peterson, A., Letarte, S., Gafken, P.R., Katz, J.E., Mallick, P., Lee, H., Schmidt, A., Ossola, R., Eng, J.K., Aebersold, R., Martin, D.B.: The standard protein mix database: a diverse dataset to assist in the production of improved peptide and protein identification software tools. *J. Proteome Res.* 7, 96–103 (2008)
16. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., Lajoie, G.: PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Commun. Mass Spectrom* 17, 2337–2342 (2003)
17. Mann, M., Wilm, M.: Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal. Chem.* 66, 4390–4399 (1994)
18. Rahman, M.S., Iliopoulos, C.S., Lee, I., Mohamed, M., Smyth, W.F.: Finding patterns with variable length gaps or don't cares. In: Chen, D.Z., Lee, D.T. (eds.) COCOON 2006. LNCS, vol. 4112, pp. 146–155. Springer, Heidelberg (2006)
19. Savidor, A., Donahoo, R., Hurtado-Gonzales, O., VerBerkmoes, N., Shah, M., Lamour, K., McDonald, W.: Expressed peptide tags: an additional layer of data for genome annotation. *J. Proteome Res.* 5, 3048–3058 (2006)
20. Tanner, S., Shu, H., Frank, A., Wang, L., Zandi, E., Mumby, M., Pevzner, P., Bafna, V.: InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.* 77, 4626–4639 (2005)



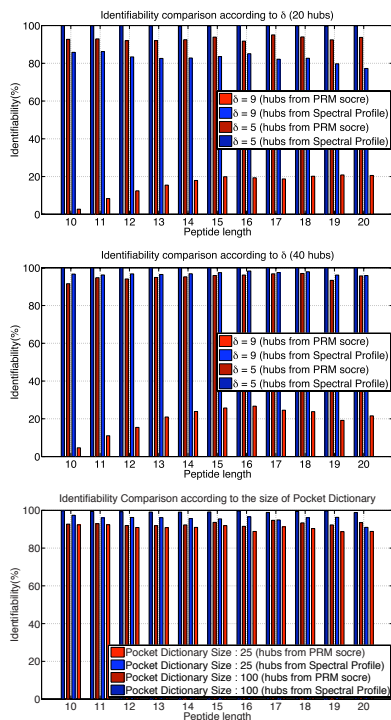
**Supplement A: Spectral Dictionary for the Peptide LNRVSQ GK****Table S1.** The Spectral Dictionary of the peptide LNRVSQ GK consists of 92 full-length peptides. The gapped peptides in the third column represent the Gapped Path Dictionary while the gapped peptides in the fourth column represent the Compact Gapped Path Dictionary.

Peptides in Spectral Dictionary	Score	Gapped peptides from Gapped Path Dictionary	Gapped peptides from Compact Gapped Path Dictionary
LNRVSQ GK	36	[227][156]VS[128]G[128]	[227][156]VS[128]G[128]
LNRVSK GK	36		
VQRVSK GK	35	[227][156]VS[128]G[128]	
VQRVSQ GK	35		
RARVSK GK	35		
NLRVSK GK	35		
ARRVSK GK	35		
RARVSQ GK	35		
VKRVSQ GK	35		
NLRVSQ GK	35		
VKRVSK GK	35		
ARRVSQ GK	35		
QRVVSQ GK	36	[128][255]VS[128]G[128]	[128][255]VS[128]G[128]
KRVVSQ GK	36		
QRVVSK GK	36		
KRVVSK GK	36		
KASPVS GK	35	[128][255]VS[128]G[128]	
QTGPVS GK	35		
QPGTVS GK	35		
QSAPVS GK	35		
KTGPVS GK	35		
KPSAVS GK	35		
KSAPVS GK	35		
QALAVS GK	35		
QASPVS GK	35		
KSAPVS GK	35		
QSPAVS GK	35		
QPSAVS GK	35		
KALAVS GK	35		
KTGPVS GK	35		
QSPAVS GK	35		
KPSAVS GK	35		
QTGPVS GK	35		
KSPAVS GK	35		
KALAVS GK	35		
KPGTVS GK	35		
QASPVS GK	35		
KASPVS GK	35		
QPSAVS GK	35		
QPGTVS GK	35		
KSPAVS GK	35		
KPGTVS GK	35		
QSAPVS GK	35		
QALAVS GK	35		

Table S1. (continued)

Peptides in Spectral Dictionary	Score	Gapped peptides from Gapped Path Dictionary	Gapped peptides from Compact Gapped Path Dictionary
QVRVSKGK	42	[128]V[156]VS[128]G[128]	[128]V[156]VS[128]G[128]
KVRVVSQGK	42		
QVRVVSQGK	42		
KVRVSKGK	42		
AGVRVSKGK	39	[128]V[156]VS[128]G[128]	
AGVRVVSQGK	39		
GAVRVSKGK	39		
GAVRVVSQGK	39		
KVGVVSKGK	37	[128]V[156]VS[128]G[128]	
KVGVVVSQGK	37		
QVVGVSQGK	37		
KVVGVSQGK	37		
QVVGVSQGK	37		
KVVGVSQGK	37		
QVGVVSQGK	37		
QVGVVSQGK	37		
KVRVVSAGGK	35	[128]V[156]VS[128]G[128]	
KVRVVSAGGK	35		
QVRVVSAGGK	35		
VRVVSAGGK	35		
QVRGEKGG	35	[128]V[156][186][128]G[128]	[128]V[156][186][128]G[128]
QVRGEQGG	35		
KVRGEQGG	35		
KVRGEKGG	35		
KVRVNTGK	36	[128]V[156]V[215]G[128]	[128]V[156]V[215]G[128]
QVRVNTGK	36		
YGYVVSQGK	36	[383]VS[128]G[128]	[383]VS[128]G[128]
YGYVSKGK	36		
QGPTVVSQGK	39	[128]G[198]VS[128]G[128]	[128]G[198]VS[128]G[128]
KGPTVVSQGK	39		
KGTPVSKGK	39		
KGPTVVSQGK	39		
QGTPVVSQGK	39		
QGPTVSKGK	39		
KGTPVVSQGK	39		
QGTPVSKGK	39		
QGVVVSQGK	37	[128]G[198]VS[128]G[128]	
KGVVVSQGK	37		
QGVVVSQGK	37		
KGVVVSQGK	37		
AGGTPVSKGK	36	[128]G[198]VS[128]G[128]	
AGGTPVSKGK	36		
GAGTPVSKGK	36		
GAGTPVVSQGK	36		
GAGTPVVSQGK	36		
GAGTPVVSQGK	36		
AGGTPVVSQGK	36		
GAGTPVSKGK	36		
AGGTPVVSQGK	36		

## Supplement B: Spectral Profile vs. PRM Score



**Fig. S1.** Comparison of hubs generated from largest Spectral Profile peaks with hubs generated from the largest PRM scores. Each hub set is evaluated by the identifiability of the resulting  $\delta$ -reduced Gapped Spectral Dictionary. (for  $\delta = 5, 9$  and the number of hubs varying from 20 (upper figure) to 40 (middle figure)). When  $\delta = 9$ , the hubs constructed from the Spectral Profile have much better identifiability than hubs constructed from PRM score (95% vs. 30%). In the bottom figure, the identifiability according to the size of Pocket Dictionary is compared (Pocket Dictionary size = 25, 100, the number of hubs is 20, and  $\delta = 5$ ). Blue bars represent the hubs from the Spectral Profile and red bars from the PRM score.

## Supplement C: Datasets

We used the *previously published* Shewanella, HEK, and Standard data sets to benchmark MS-GappedDictionary.

**Shewanella** dataset. To benchmark the performance of MS-GappedDictionary, we adopted the Shewanella dataset composed of 18,468 charge 2 spectra from *Shewanella oneidensis* MR-1, each representing a distinct tryptic peptide [13]. The spectra in this dataset were identified with InsPecT $\oplus$ MS-GeneratingFunction to ensure that all spectra have spectral probabilities below  $10^{-9}$ . Note that MS-GeneratingFunction was shown to improve upon other MS/MS identification tools (InsPecT, X!Tandem, and SEQUEST/PeptideProphet [12]) and in most applications, peptide identifications with spectral probabilities above  $10^{-9}$  are of little use since they result in high FDR.<sup>9</sup>

**HEK** dataset. The Shewanella dataset cannot be utilized to test MS-GappedDictionary in searching huge databases because *Shewanella oneidensis* MR-1 is a small genome. Kim et al. [13] used spectra from the human *HEK293* cell line and searched them against the six-frame translation of the human genome ( $\approx 2.5$  billion amino acids for repeat-masked human genome). We used the same dataset (composed of 21,605 charge 2 spectra of distinct peptides) to compare the performance of MS-GappedDictionary and MS-Dictionary in the six-frame translation of the human genome search. The spectra in this dataset were identified with InsPecT $\oplus$ MS-GeneratingFunction to ensure that all spectra have spectral probabilities below  $10^{-11}$  (see [13] for the reason to choose the rigid threshold). Each spectrum in HEK dataset is identified as a tryptic peptide from the six-frame translation of the human genome (i.e., peptides that span the exon boundaries are discarded).

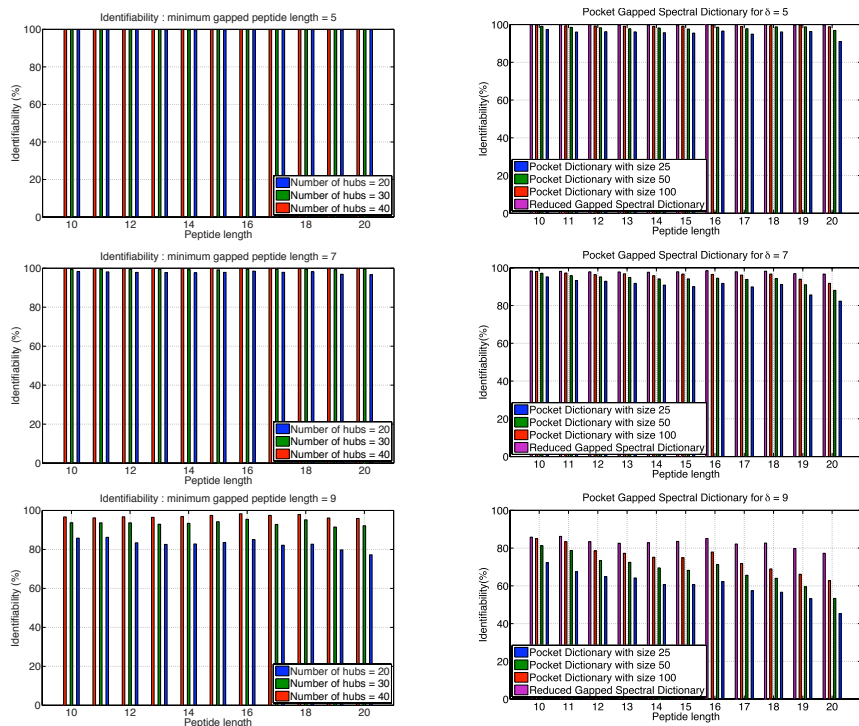
**Standard** dataset. Both Shewanella and HEK data sets are inadequate for benchmarking the (gapped) tag generation accuracy, since the tag-based tool InsPecT was used to identify the spectra (i.e., a correct InsPecT tag was generated for every spectrum). We obtained the dataset reported in [14] collected from the Standard Protein Mix database [15], where the spectra were identified by SEQUEST [3] and PeptideProphet [11] that do not use tags for identifications. We further selected peptide identifications with spectral probabilities below  $10^{-9}$  and formed the dataset (denoted *Standard*) with 990 charge 2 spectra of distinct peptides.

To generate the Gapped Spectral Dictionaries, the value of the spectral probability threshold is set to  $10^{-9}$  for Shewanella and Standard datasets and  $10^{-11}$  for HEK dataset (assuming that the precursor integer mass is known). The spectral hubs are selected based on  $k$  maximal peaks in its Spectral Profile with  $k$  varying from 20 to 40.

---

<sup>9</sup> The Supplement present analysis of the same dataset for spectral probabilities below  $10^{-10}$  and  $10^{-11}$ .

## Supplement D: Identifiability of the Gapped Spectral Dictionaries



**Fig. S2. Left panel:** Identifiability of the  $\delta$ -reduced Gapped Spectral Dictionaries from the Shewanella dataset for  $\delta = 5$  (upper part),  $\delta = 7$  (middle part), and  $\delta = 9$  (lower part).

**Right panel:** Identifiability of the  $\delta$ -reduced Gapped Spectral Dictionaries and Pocket Dictionaries from the Shewanella dataset for  $\delta = 5$  (upper part),  $\delta = 7$  (middle part), and  $\delta = 9$  (lower part). The number of hubs is 20. Even for long peptides, Pocket Dictionaries with 50 gapped peptides are sufficient to ensure the identifiability higher than 97% when  $\delta = 5$ . When  $\delta$  is large, larger Pocket Dictionaries are needed.

Supplement E: Analog of Figure 3 for Varying Number of Hubs and the Spectral Probability Thresholds

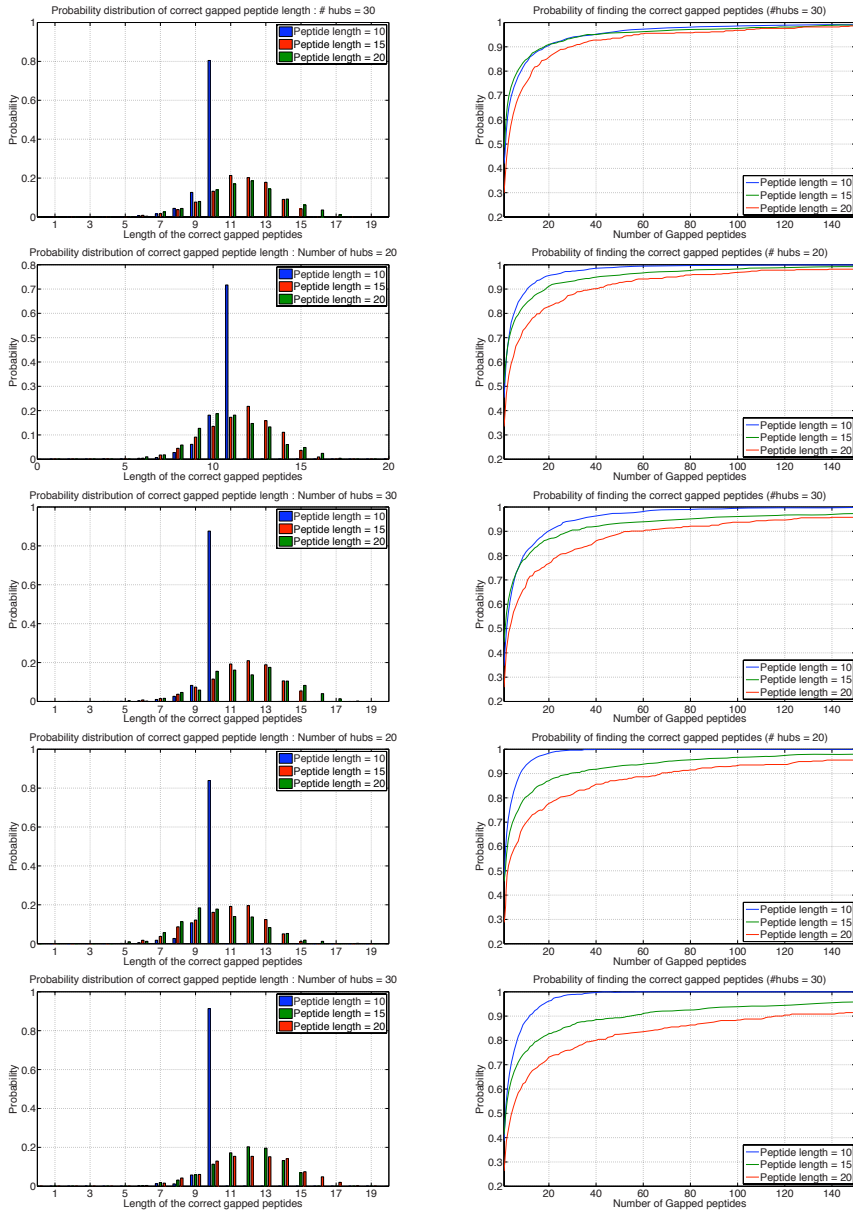
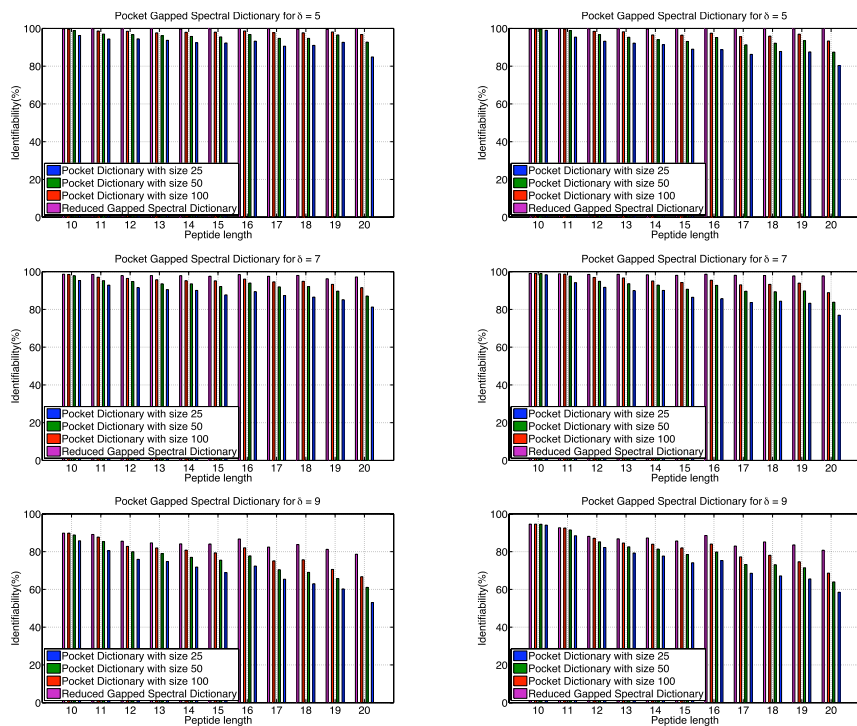


Fig. S3. Analog of Figure 3 when the number of hubs is 20 or 30 and the spectral probability threshold is  $10^{-9}$  -  $10^{-11}$  (1st row :  $10^{-9}$ , 2nd and 3rd rows :  $10^{-10}$ , 4th and 5th rows :  $10^{-11}$ ). 20 or 30 hubs are used.  $\delta$  is fixed to 5 for figures in the right panel.

## Supplement F: Spectral Identifiability for the Spectral Probability Thresholds $10^{-10}$ or $10^{-11}$



**Fig. S4. Left panel:** Analog of Figure S2 (right) when the spectral probability threshold is  $10^{-10}$ .

**Right panel:** Analog of Figure S2 (right) when the spectral probability threshold is  $10^{-11}$ .

## Supplement G: Gap Deconvolution Table

**Table S2.** Deconvolution of gaps into amino acid combinations (for integer masses from 114 Da to 250 Da)

Mass	# Comb.	Combinations
114	2	[GG, N]
128	4	[AG, GA, Q, K]
142	1	[AA]
144	2	[SG, GS]
154	2	[PG, GP]
156	3	[VG, GV, R]
158	4	[TG, SA, AS, GT]
168	2	[PA, AP]
170	4	[LG, VA, AV, GL]
171	3	[GGG, NG, GN]
172	4	[DG, TA, AT, GD]
174	1	[SS]
184	4	[LA, PS, SP, AL]
185	9	[AGG, GAG, QG, KG, GGA, NA, AN, GQ, GK]
186	7	[EG, DA, VS, SV, AD, GE, W]
188	4	[MG, TS, ST, GM]
194	3	[HG, PP, GH]
196	2	[VP, PV]
198	3	[TP, VV, PT]
199	7	[AAG, AGA, GAA, QA, KA, AQ, AK]
200	6	[EA, LS, TV, VT, SL, AE]
201	5	[SGG, GSG, GGS, NS, SN]
202	5	[MA, DS, TT, SD, AM]
204	2	[FG, GF]
208	2	[HA, AH]
210	2	[LP, PL]
211	5	[PGG, GPG, GGP, NP, PN]
212	4	[DP, LV, VL, PD]
213	8	[VGG, GVG, RG, AAA, GGV, NV, VN, GR]
214	4	[DV, LT, TL, VD]
215	15	[TGG, SAG, ASG, GTG, SGA, GSA, AGS, GAS, QS, KS, GGT, NT, TN, SQ, SK]
216	4	[ES, DT, TD, SE]
217	2	[CG, GC]
218	4	[FA, MS, SM, AF]
220	2	[YG, GY]
224	2	[HS, SH]
225	10	[PAG, APG, PGA, GPA, AGP, GAP, QP, KP, PQ, PK]
226	3	[EP, LL, PE]
227	17	[LGG, VAG, AVG, GLG, VGA, GVA, RA, AGV, GAV, QV, KV, GGL, NL, LN, VQ, VK, AR]
228	11	[GGGG, NGG, GNG, MP, EV, DL, GGN, NN, LD, VE, PM]
229	18	[DGG, TAG, ATG, GDG, TGA, SAA, ASA, GTA, AAS, AGT, GAT, QT, KT, DN, GGD, ND, TQ, TK]
230	5	[MV, ET, DD, TE, VM]
231	5	[SSG, CA, SGS, GSS, AC]
232	2	[MT, TM]
234	6	[YA, FS, HP, PH, SF, AY]
236	2	[HV, VH]
238	2	[HT, TH]
239	3	[PAA, APA, AAP]
241	19	[LAG, PSG, SPG, ALG, LGA, VAA, AVA, GLA, PGS, GPS, SGP, GSP, AAV, AGL, GAL, QL, KL, LQ, LK]
242	22	[AGGG, GAGG, QGG, KGG, GGAG, NAG, ANG, GQG, GKG, GGGA, NGA, GNA, EL, AGN, GAN, QN, KN, GGQ, NQ, GGK, NK, LE]
243	28	[EGG, DAG, VSG, SVG, ADG, GEG, WG, DGA, TAA, ATA, GDA, VGS, GVS, RS, SGV, GSV, AAT, EN, AGD, GAD, QD, KD, DQ, DK, GGE, NE, SR, GW]
244	6	[FP, ML, ED, DE, LM, PF]
245	14	[MGG, TSG, STG, GMG, SSA, TGS, SAS, ASS, GTS, SGT, GST, MN, GGM, NM]
246	4	[FV, MD, DM, VF]
247	2	[CS, SC]
248	2	[FT, TF]
250	4	[YS, HL, LH, SY]



## Supplement H: Generating Proper Gapped Tags from Gapped Peptides

We distinguish between *terminal* tags (that start at N-terminus or end at C-terminus) and *internal* tags. The tag generation algorithm attempts to generate a proper tag for each gapped peptide from the Pocket Dictionary  $\{P_1, \dots, P_n\}$  ordered in the decreasing order of (gapped) peptide coverages. At the  $i$ -th stage, the algorithm selects one proper gapped tag from peptide  $P_i$  unless (i) the peptide  $P_i$  contains one of the previously chosen proper gapped tags, or (ii) the peptide  $P_i$  does not have proper gapped tags. If there are multiple proper gapped tags available for selection at the  $i$ -th stage, the algorithm randomly selects an internal tag (if available), otherwise, it selects a terminal tag.<sup>10</sup>

Table S3 compares the percentage of gapped peptides with peptide sequence tags and proper gapped tags in Pocket Dictionaries. Table S4 shows the average numbers of gapped tags that are generated from the Pocket Dictionaries.

**Table S3.** The percentage of gapped peptides in the Pocket Dictionary of size 100 that contain peptide sequence tags and proper gapped tags of length 3 (for Standard dataset)

Peptide length	% gapped peptides with peptide sequence tags	% gapped peptides with proper gapped tags
10	90.83%	98.66%
12	69.69%	90.93%
14	54.24%	82.55%
16	47.25%	75.69%
18	35.30%	64.08%
20	38.33%	61.44%

**Table S4.** The average number of proper gapped tags (of length 3) produced by the tag generation algorithm for various peptide lengths (for Standard dataset). Only  $\approx 15$  proper gapped tags are required on average to cover the Pocket Dictionary with 100 peptides.

Peptide length	7	8	9	10	11	12	13	14	15	16	17	18	19	20
# gapped tags	1.2	6.6	13.1	15.3	16.2	17.4	17.9	16.8	17.6	18.4	16.1	16.9	17.8	15.1

<sup>10</sup> Internal proper gapped tags are preferred since they typically have better filtration efficiency than terminal tags.

### Supplement I: # of Matches of Spectra Searched against the Six Frame Translation of Human Genome before Scoring

**Table S5.** Average number of matches of spectra from HEK dataset searched against the six-frame translation of human genomes before scoring. HEK dataset is used to generate the Gapped Dictionaries. Only 3 - 60 matches should be scored to remove those that are not in the Spectral Dictionary.

Peptide length	10	11	12	13	14	15	16	17	18	19	20
# matches	2.9	9.5	24.3	38.7	54.0	44.4	55.6	53.2	54.9	53.3	66.1

# naiveBayesCall: An Efficient Model-Based Base-Calling Algorithm for High-Throughput Sequencing

Wei-Chun Kao<sup>1</sup> and Yun S. Song<sup>1,2</sup>

<sup>1</sup> Computer Science Division, University of California, Berkeley, CA 94720, USA

<sup>2</sup> Department of Statistics, University of California, Berkeley, CA 94720, USA  
wckao@eecs.berkeley.edu, yss@eecs.berkeley.edu

**Abstract.** Immense amounts of raw instrument data (i.e., images of fluorescence) are currently being generated using ultra high-throughput sequencing platforms. An important computational challenge associated with this rapid advancement is to develop efficient algorithms that can extract accurate sequence information from raw data. To address this challenge, we recently introduced a novel model-based base-calling algorithm that is fully parametric and has several advantages over previously proposed methods. Our original algorithm, called BayesCall, significantly reduced the error rate, particularly in the later cycles of a sequencing run, and also produced useful base-specific quality scores with a high discrimination ability. Unfortunately, however, BayesCall is too computationally expensive to be of broad practical use. In this paper, we build on our previous model-based approach to devise an efficient base-calling algorithm that is orders of magnitude faster than BayesCall, while still maintaining a comparably high level of accuracy. Our new algorithm is called naive-BayesCall, and it utilizes approximation and optimization methods to achieve scalability. We describe the performance of naiveBayesCall and demonstrate how improved base-calling accuracy may facilitate de novo assembly when the coverage is low to moderate.

## 1 Introduction

Recent advances in sequencing technology is enabling fast and cost-effective generation of sequence data, and complete whole-genome sequencing will soon become a routine part of biomedical research. The key feature of the next-generation sequencing technology is parallelization and the main mechanism underlying several platforms is sequencing-by-synthesis (SBS); we refer the reader to [1, 15] for a more comprehensive introduction to SBS and whole-genome re-sequencing. Briefly, tens to hundreds of millions of random DNA fragments get sequenced simultaneously by sequentially building up complementary bases of single-stranded DNA templates and by capturing the synthesis information in a series of raw images of fluorescence. Extracting the actual sequence information (i.e., strings in  $\{A, C, G, T\}$ ) from image data involves two computational problems, namely image analysis and base-calling. The primary function of image

analysis is to translate image data into fluorescence intensity data for each DNA fragment, while the goal of base-calling is to infer sequence information from the obtained intensity data. Although algorithms developed by the manufacturers of the next-generation sequencing platforms work reasonably well, it is widely recognized that independent researchers must develop improved algorithms for optimizing data acquisition, to reduce the error rate and to reduce the cost of sequencing by increasing the throughput per run.

At present, Illumina's Genome Analyzer (GA) is the most widely-used system among the competing next-generation sequencing platforms. In GA, SBS is carried out on a glass surface called the flow cell, which consists of 8 lanes, each with 100 tiles. In a typical sequencing run, each tile holds about a hundred thousand clusters, with each cluster containing about 1000 identical DNA templates. The overall objective is to infer the sequence information for each cluster. The base-calling software supplied with GA is called Bustard, which adopts a very efficient algorithm based on matrix inversion. Although the algorithm works very well for the early cycles of a sequencing run, it is well-known that the error rate of Bustard becomes substantial in later cycles. Reducing the error rate of base-calls and improving the accuracy of base-specific quality score will have important practical implications for assembly [3, 4, 11, 12, 14, 17, 21], polymorphism detection (especially rare ones) [2, 12], and downstream population genomics analysis of next-generation sequencing data [7, 8].

Recently, several improved base-calling algorithms [5, 9, 16, 19] have been developed for the Illumina platform. In particular, a large improvement in accuracy was achieved by our own method called BayesCall [9]. The key feature that distinguishes BayesCall from the other methods is the explicit modeling of the sequencing process. In particular, BayesCall explicitly takes residual effects into account and is the only existing base-calling algorithm that can incorporate time-dependent parameters. Importantly, parameter estimation is done *unsupervised* and BayesCall produces very good results even when using a very small training set consisting of only a few hundred randomly chosen clusters. This feature enables the estimation of local parameters to account for the potential differences between different tiles and lanes. Furthermore, being a fully parametric model, our approach provides information on the relative importance of various factors that contribute to the observed intensities, and such information may become useful for designing an improved sequencing technology.

*Supervised* machine learning is an alternative approach that other researchers have considered in the past for base-calling. For example, Alta-Cyclic [5] is a method based on the support vector machine that requires a large amount of labeled training data. To create a rich training library, in every sequencing run it requires using a control lane containing a sample with a known reference genome. Note that using such a control incurs cost and takes up space on the flow cell that could otherwise be used to sequence a sample of interest to the biologist. Furthermore, this approach cannot handle variability across lanes.

In [9], we showed that our method significantly improves the accuracy of base-calls, particularly in the later cycles of a sequencing run. In addition, we showed

that BayesCall produces quality scores with a high discrimination ability [6] that consistently outperforms both Bustard’s and Alta-Cyclic’s. Unfortunately, however, this improvement in accuracy came at the price of substantial increase in running time. BayesCall is based on a generative model and performs base-calls by maximizing the posterior distribution of sequences given observed data (i.e., fluorescence intensities). This step involves using the Metropolis-Hastings algorithm with simulated annealing, which is computationally expensive; it would take several days to base-call a single lane using a desktop computer. This slow running time seriously restricts the practicality of BayesCall.

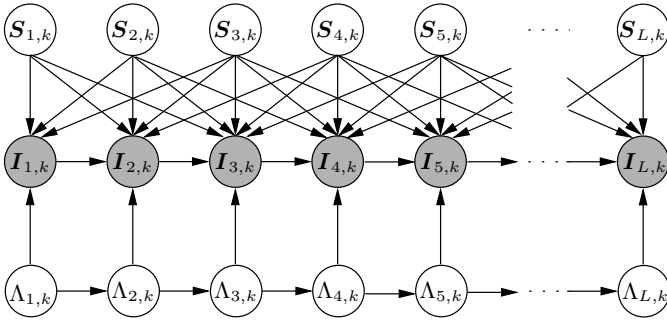
The goal of this paper is to build on the ideas behind BayesCall to devise an efficient base-calling algorithm that is orders of magnitude faster than BayesCall, while still maintaining a comparably high level of accuracy. There are two computational parts to BayesCall: parameter estimation and base-calling. Since estimation of the time-dependent parameters in BayesCall can be performed progressively as the sequencing machine runs, we believe that the bottleneck is in the base-calling part. Our new algorithm is called naiveBayesCall. It is based on the same generative model as in BayesCall and employs the same parameter estimation method as before (see [9] for details). However, in contrast to BayesCall, our new algorithm avoids doing Markov chain Monte Carlo sampling in the base-calling part of the algorithm. Instead, naiveBayesCall utilizes approximation and optimization methods to achieve scalability. To test the performance of our method, we use a standard resequencing data of PhiX174 virus, obtained from a 76-cycle run on Illumina’s GA II platform. Then, we demonstrate how improved base-calling accuracy may facilitate *de novo* assembly.

Our software implementation can be run either on an ordinary PC or on a computing cluster, and is fully compatible with the file formats used by Illumina’s GA pipeline. Our software is available at <http://bayescall.sourceforge.net/>.

## 2 A Review of the Model Underlying the Original BayesCall Algorithm

Our main goal in developing BayesCall was to model the sequencing process in GA to the best of our knowledge, by taking stochasticity into account and by explicitly modeling how errors may arise. In each cycle, ideally the synthesis process is supposed to add exactly one complementary base to each template, but, unfortunately, this process is not perfect and some templates may jump ahead (called prephasing) or lag behind (called phasing) in building up complementary strands. This is a major source of complication for base-calling. Environment factors such as temperature fluctuation also contribute to stochasticity.

Below, we briefly review the main ideas underlying BayesCall [9]. Throughout, we adopt the same notational convention as in [9]: Multi-dimensional variables are written in boldface, while scalar variables are written in normal face. The transpose of a matrix  $\mathbf{M}$  is denoted by  $\mathbf{M}'$ . The index  $t$  is used to refer to a particular cycle, while the index  $k$  is used to refer to a particular cluster of identical DNA templates. The total number of cycles in a run is denoted by  $L$ .



**Fig. 1.** The graphical model for BayesCall. The observed random variables are the intensities  $I_{t,k}$ . Base-calling is done by finding the maximum a posteriori estimates of  $S_{t,k}$ . In this illustration, the window within which we consider phasing and prephasing effects has size 5. In our implementation, we use a window of size 11.

Let  $e_i$  denote a 4-component column unit vector with a 1 in the  $i$ th entry and 0s elsewhere. We use the basis with A, C, G, T corresponding to indices 1, 2, 3, 4, respectively.

BayesCall is founded on a graphical model, illustrated in Figure 1. It involves the following random variables:

**Sequence ( $S_k$ ):** We use  $S_k = (S_{1,k}, \dots, S_{L,k})$ , with  $S_{t,k} \in \{e_A, e_C, e_G, e_T\}$ , to denote the 4-by- $L$  binary *sequence matrix* corresponding to the complementary sequence of the DNA templates in cluster  $k$ . The main goal of base-calling is to infer  $S_k$  for each cluster  $k$ . We assume a uniform prior on sequences:

$$S_{t,k} \sim \text{Unif}(e_A, e_C, e_G, e_T).$$

If the genome-wide nucleotide distribution of the sample is known, then that distribution may be used here instead, which should improve the accuracy of base-calls.

**Active template density ( $A_{t,k}$ ):** In BayesCall, fluctuation in fluorescence intensity over time is explicitly modeled using a random variable  $A_{t,k}$  that corresponds to the per-cluster density of templates that are “active” (i.e., able to synthesize further) at cycle  $t$  in cluster  $k$ . Given  $A_{t-1,k}$  from the previous cycle,  $A_{t,k}$  is distributed as a 1-dimensional normal distribution with mean  $(1 - d_t)A_{t-1,k}$  and variance  $(1 - d_t)^2 A_{t-1,k}^2 \sigma_t^2$ :

$$A_{t,k} | A_{t-1,k} \sim \mathcal{N}((1 - d_t)A_{t-1,k}, (1 - d_t)^2 A_{t-1,k}^2 \sigma_t^2). \tag{1}$$

**Intensities ( $I_{t,k}$ ):** We use  $I_{t,k} = (I_{t,k}^A, I_{t,k}^C, I_{t,k}^G, I_{t,k}^T)' \in \mathbb{R}^{4 \times 1}$  to denote the fluorescence intensities of A, C, G, T channels at cycle  $t$  in cluster  $k$ , after subtracting out the background signals. These are the observed random variables in our graphical model.

As mentioned above, the DNA synthesis process is not perfect and may go out of “phase.” In Bustard and BayesCall, the synthesis process in a cycle is modeled by a Markov model in which the position of the terminating complementary nucleotide of a given template changes from  $i$  to  $j$  according to the transition matrix  $\mathbf{P} = (P_{ij})$  given by

$$P_{ij} = \begin{cases} p, & \text{if } j = i, \\ 1 - p - q, & \text{if } j = i + 1, \\ q, & \text{if } j = i + 2, \\ 0, & \text{otherwise,} \end{cases}$$

where  $0 \leq i, j \leq L$ . Here,  $p$  denotes the probability of phasing (i.e., no new base is synthesized during the cycle), while  $q$  denotes the probability of prephasing (i.e., two bases are synthesized). Normal synthesis of a single complementary nucleotide occurs with probability  $1 - p - q$ . At cycle 0, we assume that all templates start at position 0; i.e., no nucleotide has been synthesized. Note that the  $(i, j)$  entry of the matrix  $\mathbf{P}^t$  corresponds to the probability that a terminator at position  $i$  moves to position  $j$  after  $t$  cycles.

Define  $Q_{jt}$  as the probability that a template terminates at position  $j$  after  $t$  cycles. It is easy to see that  $Q_{jt} = [\mathbf{P}^t]_{0,j}$ , the  $(0, j)$  entry of the matrix  $\mathbf{P}^t$ . We use  $\mathbf{Q}_t$  to denote column  $t$  of the  $L$ -by- $L$  matrix  $\mathbf{Q} = (Q_{jt})$ . In practice,  $\mathbf{Q}_t$  will have only a few dominant components, with the rest being very small. More precisely, the dominant components will be concentrated about the  $t$ th entry. Therefore, at cycle  $t$ , we simplify the computation by considering phasing and prephasing effects only within a small window  $w$  about position  $t$ . Let  $\mathbf{Q}_t^w$  denote the  $L$ -dimensional column vector obtained from  $\mathbf{Q}_t$  by setting the entries outside the window to zero. Hence, the concentration of active templates in cluster  $k$  with A, C, G, T terminating complementary nucleotide can be approximated by the following 4-dimensional vector:

$$\mathbf{Z}_{t,k}^w = \Lambda_{t,k} \mathbf{S}_k \mathbf{Q}_t^w. \quad (2)$$

The four fluorophores used to distinguish different terminating nucleotides have overlapping spectra [20], and this effect can be modeled as  $\mathbf{X}_t \mathbf{Z}_{t,k}^w$ , where  $\mathbf{X}_t \in \mathbb{R}^{4 \times 4}$  is a matrix called the *crossstalk* matrix, with  $(X_t)_{ij}$  denoting the response in channel  $i$  due to fluorescence of a unit concentration of base  $j$ . (We refer the reader to [13] for discussion on estimating  $\mathbf{X}_t$ .)

In addition to phasing and prephasing effects, we observed other residual effects that propagate from one cycle to the next. We found that modeling such extra residual effects improves the base-call accuracy. In BayesCall, we introduced parameters  $\alpha_t$  and assumed that the observed intensity  $\mathbf{I}_{t,k}$  at cycle  $t$  contains the residual contribution  $\alpha_t(1 - d_t)\mathbf{I}_{t-1,k}$  from the previous cycle. In summary, the mean fluorescence intensity for cluster  $k$  at cycle  $t$  is given by

$$\boldsymbol{\mu}_{t,k} = \mathbf{X}_t \mathbf{Z}_{t,k}^w + \alpha_t(1 - d_t)\mathbf{I}_{t-1,k}, \quad (3)$$

with  $\mathbf{I}_{0,k}$  defined as the zero vector  $\mathbf{0}$ . Finally, with the assumption that the background noise at cycle  $t$  is distributed as Gaussian white noise with zero

mean and covariance matrix  $\|\mathbf{Z}_{t,k}^w\|^2 \boldsymbol{\Sigma}_t$ , where  $\|\cdot\|$  denotes the 2-norm, the observed fluorescence intensity in BayesCall is distributed as the following 4-dimensional normal distribution:

$$\mathbf{I}_{t,k} | \mathbf{I}_{t-1,k}, \mathbf{S}_k, A_{t,k} \sim \mathcal{N}(\boldsymbol{\mu}_{t,k}, \|\mathbf{Z}_{t,k}^w\|^2 \boldsymbol{\Sigma}_t), \tag{4}$$

where the mean is shown in (3).

In BayesCall, global parameters  $p, q$ , and cycle-dependent parameters  $d_t, \alpha_t, \sigma_t, \mathbf{X}_t, \boldsymbol{\Sigma}_t$  are estimated using the expectation-maximization (EM) algorithm, combined with Monte-Carlo integration via the Metropolis-Hastings algorithm.

### 3 naiveBayesCall: A New Algorithm

We now describe our new algorithm naiveBayesCall. As mentioned in Introduction, it is based on the same graphical model as in BayesCall, and we employ the method detailed in [9] to estimate the parameters in the model. The main novelty of naiveBayesCall is in the base-calling part of the method. We divide the presentation of our new base-calling algorithm into two parts. First, we propose a hybrid algorithm that combines the model described in Section 2 with the matrix inversion approach employed in Bustard. Then, we use the hybrid algorithm to initialize an optimization procedure that both improves the base-call accuracy and produces useful per-base quality scores.

#### 3.1 A Hybrid Base-Calling Algorithm

We present a new inference algorithm for the model described in Section 2. The main strategy is to avoid direct inference of the continuous random variables  $A_{t,k}$ . First, for each cycle  $t$ , we estimate the average concentration  $c_t$  of templates within each tile. In [9], we showed that the magnitude of the fluctuation rate  $d_t$  (c.f., (1)) is typically very small (less than 0.03) for all  $1 \leq t \leq L$ . Hence, assuming that  $d_t$  is close to zero for all  $t$ , we estimate the tile-wide average concentration  $c_t$  using

$$c_t = \frac{1}{K} \sum_{k=1}^K \sum_{b=1}^4 \max(0, [\mathbf{X}_t^{-1}(\mathbf{I}_{t,k} - \alpha_t \mathbf{I}_{t-1,k})]_b), \tag{5}$$

where  $K$  denotes the total number of clusters in the tile and  $[\mathbf{y}]_b$  denotes the  $b$ th component of vector  $\mathbf{y}$ . The above  $c_t$  serves as our estimate of  $A_{t,k}$  for all clusters  $k$  within the same tile. Using this estimate, we define

$$\tilde{\mathbf{I}}_{t,k} = \left( \mathbf{I}_{t,k} - \alpha_t \frac{c_t}{c_{t-1}} \mathbf{I}_{t-1,k} \right)_+, \tag{6}$$

where  $(\mathbf{y})_+$  denotes the vector obtained from  $\mathbf{y}$  by replacing all negative components with zeros. Note that subtracting  $\alpha_t \frac{c_t}{c_{t-1}} \mathbf{I}_{t-1,k}$  from  $\mathbf{I}_{t,k}$  accounts for the residual effect modeled in (3). The ratio  $\frac{c_t}{c_{t-1}}$  rescales  $\mathbf{I}_{t-1,k}$  so that its norm is similar to that of  $\mathbf{I}_{t,k}$ .



**Algorithm 1.** Hybrid Algorithm

---

```

for all tiles do
  for all cycles  $1 \leq t \leq L$  do
    Estimate concentration  $c_t$  for each cycle  $t$  according to (5).
  end for
  for all clusters  $1 \leq k \leq K$  do
    Compute residual-corrected intensities  $\tilde{\mathbf{I}}_k$  using (6).
    Compute concentration matrix  $\mathbf{Z}_k$  according to (7).
    Correct for phasing and prephasing effect using (8).
    Infer  $\mathbf{S}_{1,k}^H, \dots, \mathbf{S}_{L,k}^H$  using (9) and output the associated sequence.
  end for
end for

```

---

After determining  $\tilde{\mathbf{I}}_{t,k}$ , the rest of the hybrid base-calling algorithm resembles Bustard. (For a detailed description of Bustard, see [9].) First, for each cycle  $t$ , we estimate the cluster-specific normalized concentration of four different bases using

$$\mathbf{z}_{t,k} = (\mathbf{z}_{t,k}^A, \mathbf{z}_{t,k}^C, \mathbf{z}_{t,k}^G, \mathbf{z}_{t,k}^T)' = \frac{1}{c_t} (\mathbf{X}_t^{-1} \tilde{\mathbf{I}}_{t,k})_+, \quad (7)$$

where  $\mathbf{X}_t$  is the 4-by-4 crosstalk matrix at cycle  $t$  (see previous section). Normalizing by the tile-wide average  $c_t$  is to make the total concentration stay roughly the same across all cycles. Note that  $\mathbf{z}_{t,k}$  is an estimate of the concentration vector shown in (2). Now, we let  $\mathbf{z}_k = (\mathbf{z}_{1,k}, \dots, \mathbf{z}_{L,k})$  and use the following formula to correct for phasing and prephasing effects:

$$\mathbf{z}_k (\mathbf{Q}^w)^{-1}, \quad (8)$$

where  $\mathbf{Q}^w = (\mathbf{Q}_1^w, \dots, \mathbf{Q}_L^w)$  is the  $L$ -by- $L$  phasing-prephasing matrix defined in Section 2. Finally, for  $t = 1, \dots, L$ , the row index of the largest value in column  $t$  of (8) is called as the  $t$ th base of the DNA templates in cluster  $k$ :

$$\mathbf{S}_{t,k}^H = \operatorname{argmax}_{b \in \{A,C,G,T\}} [\mathbf{z}_k (\mathbf{Q}^w)^{-1}]_{b,t}. \quad (9)$$

Algorithm 1 summarizes the hybrid base-calling algorithm just described.

The performance of the hybrid algorithm will be discussed in Section 4. We will see that, with the parameters estimated in BayesCall, our simple hybrid algorithm already outperforms Bustard.

### 3.2 Estimating $\Lambda_k$ via Optimization and Computing Quality Scores

In this section, we devise a method to improve the hybrid algorithm described above and to compute base-specific quality score. The Viterbi algorithm [18] has been widely adopted as a dynamic programming algorithm to find the most probable path of states in a hidden Markov Model. There are two source of difficulty in applying the Viterbi algorithm to our problem:

---

**Algorithm 2.** naiveBayesCall Algorithm

---

```

for all clusters  $k$  do
  Initialize  $\mathbf{S}_k^{(0)} = (\mathbf{S}_{1,k}^H, \dots, \mathbf{S}_{L,k}^H)$  using Algorithm 1
  for  $1 \leq t \leq L$  do
    for  $b \in \{A, C, G, T\}$  do
      Find  $\lambda_{t,k}^b = \operatorname{argmax}_{\lambda} L_{t,k}^b(\lambda)$ , where  $L_{t,k}^b(\lambda)$  is defined as in (10).
      Compute base-specific quality score  $Q(b)$  using (14) and (15).
    end for
    Set  $s_{t,k} = \operatorname{argmax}_{b \in \{A, C, G, T\}} L_{t,k}^b(\lambda_{t,k}^b)$ .
    Update  $\mathbf{S}_k^{(t)} = \mathcal{R}_{t,s_{t,k}}(\mathbf{S}_k^{(t-1)})$ .
  end for
  Call  $s_{1,k}, \dots, s_{L,k}$  as the inferred sequence and output base-specific quality scores.
end for

```

---

1. Our model is a high order Markov model, so path tracing can be computationally expensive. This complexity arises from modeling phasing and prephasing effects. Recall that the observation probability at a given cycle  $t$  depends on all hidden random variables  $\mathbf{S}_{i,k}$  with  $i$  within a window  $w$  about  $t$ . In [9], we used 11 for the window size.
2. In addition to the discrete random variables  $\mathbf{S}_k = (\mathbf{S}_{1,k}, \dots, \mathbf{S}_{L,k})$  for the DNA sequence, our model contains continuous hidden random variables  $\mathbf{A}_k = (A_{1,k}, \dots, A_{L,k})$ , but the Viterbi algorithm cannot handle continuous variables. One might try to address this problem by marginalizing out  $\mathbf{A}_k$ , but it turns out that the maximum a posteriori (MAP) estimate of  $\mathbf{A}_k$  is useful for computing quality scores.

To address the first problem, we obtain a good initial guess of hidden variables  $\mathbf{S}_k$  and use it to break the high order dependency. To cope with the second problem, we adopt a sequential approach. Algorithm 2 summarizes our naiveBayesCall algorithm and a detailed description is provided below.

Our algorithm iteratively estimates  $\Lambda_{t,k}$  and updates  $\mathbf{S}_{t,k}$ , starting with  $t = 1$  and ending at  $t = L$ . Let  $\mathbf{S}_k^{(i)}$  denote the sequence matrix after the  $i$ th iteration. We initialize  $\mathbf{S}_k^{(0)} = \mathbf{S}_k^H$ , where  $\mathbf{S}_k^H = (\mathbf{S}_{1,k}^H, \dots, \mathbf{S}_{L,k}^H)$  is obtained using the hybrid algorithm described in Section 3.1. Let  $s_{t,k}$  denote the base (i.e., A, C, G, or T) called by naiveBayesCall for position  $t$  of the DNA sequence in cluster  $k$ . At iteration  $t$ , the first  $t - 1$  bases  $s_{1,k}, \dots, s_{t-1,k}$  have been called and the vectors  $\mathbf{S}_{1,k}, \dots, \mathbf{S}_{t-1,k}$  have been updated accordingly. The following procedures are performed at iteration  $t$ :

**Optimization:** Our inference of  $\Lambda_{t,k}$  depends on the base at position  $t$ , which has not been called yet. We use  $\lambda_{t,k}^b$  to denote the inferred value of  $\Lambda_{t,k}$ , given that the base at position  $t$  is  $b$ . For a given base  $b \in \{A, C, G, T\}$ , we define the log-likelihood function

$$L_{t,k}^b(\lambda) = \begin{cases} \log \mathbb{P}[\mathbf{I}_{t,k} | \mathbf{I}_{t-1,k}, \mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)}), \lambda], & \text{if } t = 1, \\ \log \mathbb{P}[\lambda | \lambda_{t-1,k}^{s_{t-1,k}}] + \log \mathbb{P}[\mathbf{I}_{t,k} | \mathbf{I}_{t-1,k}, \mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)}), \lambda], & \text{if } t > 1, \end{cases} \quad (10)$$

where  $\mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)})$  denotes the sequence matrix obtained by replacing column  $t$  of  $\mathbf{S}_k^{(t-1)}$  with the unit column vector  $\mathbf{e}_b$ , the probability  $\mathbb{P}[\lambda | \lambda_{t-1,k}^{s_{t-1,k}}]$  is defined in (11), and observation likelihood  $\mathbb{P}[\mathbf{I}_{t,k} | \mathbf{I}_{t-1,k}, \mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)})]$  is defined by (12)–(14). More exactly,

$$\mathbb{P}[\mathbf{I}_{t,k} | \mathbf{I}_{t-1,k}, \mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)})] \approx \phi(\mathbf{I}_{t,k}; \lambda \mathbf{X}_t \mathbf{z}_{t,k}^{w,b} + \alpha_t (1 - d_t) \mathbf{I}_{t-1,k}, \|\lambda \mathbf{z}_{t,k}^{w,b}\|^2 \boldsymbol{\Sigma}_t), \quad (11)$$

where  $\mathbf{z}_{t,k}^{w,b} = \mathcal{R}_{t,b}(\mathbf{S}_k^{(t-1)}) \mathbf{Q}_t^w$  is an unscaled concentration vector and  $\phi(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$  is the probability density function of a multivariate normal distribution with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ . For each  $b \in \{A, C, G, T\}$ , we estimate  $\lambda_{t,k}^b$  using the following optimization:

$$\lambda_{t,k}^b = \underset{\lambda}{\operatorname{argmax}} L_{t,k}^b(\lambda). \quad (12)$$

Our implementation of naiveBayesCall uses the golden section search method (10) to solve the 1-dimension optimization problem in (12).

**Base-calling:** The nucleotide at position  $t$  is called as

$$s_{t,k} = \underset{b \in \{A, C, G, T\}}{\operatorname{argmax}} \max_{\lambda} L_{t,k}^b(\lambda) = \underset{b \in \{A, C, G, T\}}{\operatorname{argmax}} L_{t,k}^b(\lambda_{t,k}^b), \quad (13)$$

and the sequence matrix is updated accordingly:  $\mathbf{S}_k^{(t)} = \mathcal{R}_{t,s_{t,k}}(\mathbf{S}_k^{(t-1)})$ .

**Quality score:** For position  $t$ , the probability of observing  $b$  is estimated by

$$\mathbb{P}(b) = \frac{\phi(\mathbf{I}_{t,k}; \lambda_{t,k}^b \mathbf{X}_t \mathbf{z}_{t,k}^{w,b} + \alpha_t (1 - d_t) \mathbf{I}_{t-1,k}, \|\lambda_{t,k}^b \mathbf{z}_{t,k}^{w,b}\|^2 \boldsymbol{\Sigma}_t)}{\sum_{x \in \{A, C, G, T\}} \phi(\mathbf{I}_{t,k}; \lambda_{t,k}^x \mathbf{X}_t \mathbf{z}_{t,k}^{w,x} + \alpha_t (1 - d_t) \mathbf{I}_{t-1,k}, \|\lambda_{t,k}^x \mathbf{z}_{t,k}^{w,x}\|^2 \boldsymbol{\Sigma}_t)}, \quad (14)$$

and the quality score for base  $b$  is given by

$$Q(b) = 10 \log_{10} \left[ \frac{\mathbb{P}(b)}{1 - \mathbb{P}(b)} \right]. \quad (15)$$

## 4 Results

In this section, we compare the performance of our new algorithm naiveBayesCall with that of Bustard, Alta-Cyclic (5), and our original algorithm BayesCall (9).

### 4.1 Data and Test Setup

In our empirical study, we used a standard resequencing data of PhiX174 virus, provided to us by the DPGP Sequencing Lab at UC Davis. The data were obtained from a 76-cycle run on the Genome Analyzer II platform, with the viral sample in a single lane of the flow cell. The lane consisted of 100 tiles, containing a total of 14,820,478 clusters. Illumina’s base-calling pipeline, called

Integrated Primary Analysis and Reporting, was applied to the image data to generate intensity files.

The entire intensity data were used to train Alta-Cyclic and BayesCall. Further, since Alta-Cyclic requires a labeled training set, the reads base-called by Bustard and the PhiX174 reference genome were also provided to Alta-Cyclic. To estimate parameters in BayesCall and naiveBayesCall, the intensity data for only 250 randomly chosen clusters were used.

To create a classification data set for testing the accuracy of the four base-calling algorithms, the sequences base-called by Bustard were aligned against the PhiX174 reference genome, and those reads containing more than 22 mismatches (i.e., with more than 30% of difference) were discarded. This filtering step reduced the total number of clusters to 6,855,280, and the true sequence associated with each cluster was assumed to be the 76-bp string in the reference genome onto which the alignment algorithm mapped the sequence base-called by Bustard. The same set of clusters was used to test the accuracy of all four methods.

Note that since the classification data set was created by dropping those clusters for which Bustard produced many errors, the above experiment setup slightly favored Bustard. Also, it should be pointed out that since Alta-Cyclic was trained on the entire lane, it actually had access to the entire testing data set during the training phase.

## 4.2 Improvement in Running Time

The experiments were done on a Mac Pro with two quad-core 3.0GHz Intel Xeon processors, utilizing all eight cores. Table I(a) shows the training time and the prediction time of Alta-Cyclic, BayesCall, and naiveBayesCall. The times reported in Table I(a) are for the full-lane of data. The training time of naiveBayesCall is the same as that of BayesCall, since naiveBayesCall currently uses the same parameter estimation method as in BayesCall. Although the training time of BayesCall is longer than that of Alta-Cyclic, we point out that, in principle, the cycle-dependent parameters in BayesCall can be estimated progressively as the sequencing machine runs (a run currently takes about 10 days). This advantage comes from the fact that BayesCall can be trained without labeled training data. As Table I(a) illustrates, naiveBayesCall dramatically improves the base-calling time over BayesCall, delivering about 60X speedup. This improvement makes our model-based base-calling approach practical.

## 4.3 Summary of Base-Call Accuracy

Table I(b) shows the overall base-call accuracy of the four different methods. The columns under the label “4 Tiles” show the results for only 4 out of the 100 tiles in the lane. Since it would take more than 15 days for BayesCall to call bases for the entire lane, it was not used in the full-lane study. Both Bustard and Alta-Cyclic were trained on the full-lane data. To train BayesCall for the 4-tile data, we randomly chose 250 clusters from each tile to estimate tile-specific parameters,

**Table 1.** Comparison of overall performance results (a) Running times (in hours). (b) Base-call error rates. BayesCall’s testing time was estimated from that for 4 tiles of data. The “by-base” error rate refers to the ratio of the number of miscalled bases to the total number of base-calls made, while the “by-read” error rate refers to the ratio of the number of reads each with at least one miscalled base to the total number of reads considered.

(a)

	Training Time	Testing Time for Full-Lane
Alta-Cyclic	10	4.4
BayesCall	19	362.5
naiveBayesCall	19	6

(b)

	4 Tiles		Full-Lane	
	By-base	By-read	By-base	By-read
Bustard	0.0098	0.2656	0.0103	0.2705
Alta-Cyclic	0.0097	0.3115	0.0101	0.3150
BayesCall	0.0076	0.2319	NA	NA
naiveBayesCall	0.0080	0.2348	0.0088	0.2499

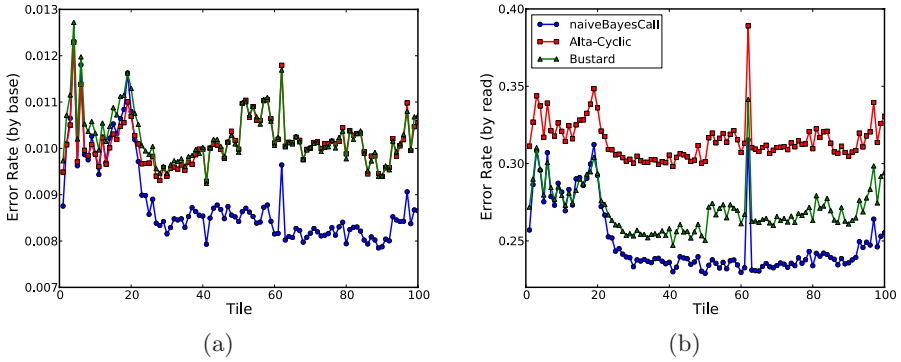
and used the same parameters in naiveBayesCall. To run naiveBayesCall on the full-lane data, we randomly chose 250 clusters from the entire lane to estimate lane-wide parameters.

From Table 1(b), we see that the performance of naiveBayesCall is comparable to that of BayesCall. Figure 2 shows the tile-specific average error rate for each tile of the full-lane data. Note that naiveBayesCall clearly outperforms both Bustard and Alta-Cyclic for tiles 21 to 100, but has comparable error rates for tiles 1 to 20. It is possible to improve naiveBayesCall’s accuracy for the first 20 tiles by using tile-specific parameter estimates (see Discussion).

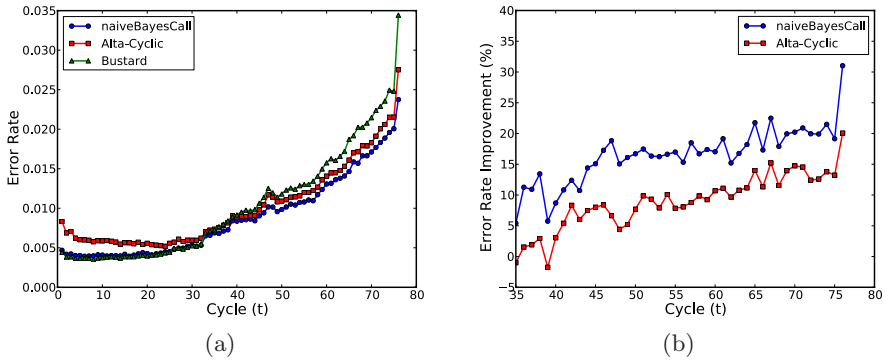
Figure 3(a) illustrates the cycle-specific average error rate. Note that naiveBayesCall’s average accuracy dominates Alta-Cyclic’s for all cycles. Furthermore, the improvement of naiveBayesCall over Bustard increases with cycles, as illustrated in Figure 3(b). This suggests that it is possible to run the sequencing machine for longer cycles and still obtain useful sequence information for longer reads by using an improved base-calling algorithm such as ours. Furthermore, we believe that fewer errors in later cycles may facilitate de novo assembly. We return to this point in Section 4.5.

#### 4.4 Discrimination Ability of Quality Scores

To compare the utility of quality scores, we follow the idea in [6] and define the discrimination ability  $D(\epsilon)$  at error tolerance  $\epsilon$  as follows. First sort the called bases according to their quality scores in decreasing order. Then go down that sorted list until the error rate surpasses  $\epsilon$ . The number of correctly called bases up to this point is defined as  $D(\epsilon)$ . Hence,  $D(\epsilon)$  corresponds to the number of

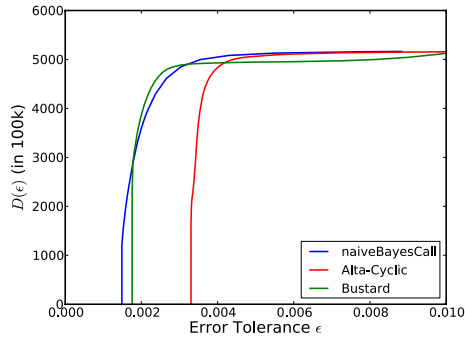


**Fig. 2.** Title-specific error rates. Our algorithm naiveBayesCall clearly outperforms both Bustard and Alta-Cyclic for tiles 21 to 100, but has comparable error rates for tiles 1 to 20. It is possible to improve naiveBayesCall’s base-call accuracy for the first 20 tiles by using tile-specific parameter estimates. (a) By-base error rate for each tile. (b) By-read error rate for each tile.



**Fig. 3.** Comparison of average base-call accuracy for the full-lane data. Note that naiveBayesCall’s average accuracy dominates Alta-Cyclic’s for all cycles. Further, The improvement of naiveBayesCall over Bustard increases with cycles. (a) Cycle-specific error rate. (b) Improvement of naiveBayesCall and Alta-Cyclic in cycle-specific error rate over Bustard.

bases that can be correctly called at error tolerance  $\epsilon$ , if we use quality scores to discriminate bases with lower error probabilities from those with higher error probabilities. For any given  $\epsilon$ , a good quality score should have a high  $D(\epsilon)$ . Shown in Figure 4 is a plot of  $D(\epsilon)$  for naiveBayesCall, Alta-Cyclic, and Bustard. As the figure shows, naiveBayesCall’s quality score consistently outperforms Alta-Cyclic’s. For  $\epsilon < 0.0017$  and  $\epsilon > 0.0032$ , naiveBayesCall’s quality score has a higher discrimination ability than Bustard’s, while the opposite is true for the intermediate values  $0.0017 < \epsilon < 0.0032$ .



**Fig. 4.** Discrimination ability  $D(\epsilon)$  of quality scores for the full-lane data

#### 4.5 Effect of Base-Calling Accuracy on the Performance of de Novo Assembly

Here, we demonstrate how improved base-calling accuracy may facilitate de novo assembly. Because of the short read length and high sequencing error rate, de novo assembly of the next-generation sequencing data is a challenging task. Recently, several promising algorithms [3, 4, 14, 17, 21] have been proposed to tackle this problem. In our study, we used the program Velvet [21] to perform de novo assembly of the reads called by different base-calling algorithms. First, we randomly chose a set of clusters from the 4-tile data without doing any filtering. Then, we base-called those clusters using each of Bustard, Alta-Cyclic, BayesCall, and naive-BayesCall, producing four different sets of base-calls on the same data set. For each set of base-called reads, Velvet was run with the  $k$ -mer length set to 55. For a given choice of coverage, we repeated this experiment 100 times. The results are summarized in Table 2, which shows the N50 length, the maximum contig length, and the total number of contigs produced; these numbers were averaged over the 100 experiments. On average naiveBayesCall led to better de novo assemblies than did Bustard or Alta-Cyclic: For 5X and 10X coverages, the performance of naiveBayesCall was similar to that of Bustard's in terms of the N50 and maximum contig lengths, but naiveBayesCall produced significantly more contigs than did Bustard. For 15X and 20X, naive-BayesCall clearly outperformed Bustard in all measures, producing longer and more contigs. The results for BayesCall and naiveBayesCall were comparable.

## 5 Discussion

Reducing the base-call error rate has important consequences for several subsequent computational problems, including assembly, SNP calling, disease association mapping, and population genomics analysis. In this paper, we have developed new algorithms to make our model-based base-calling approach scalable. Being a fully-parametric model, our approach is transparent and provides quantitative insight into the underlying sequencing process. The improvement

**Table 2.** Average contig lengths resulting from de novo assembly of the 76-cycle PhiX174 data, when different base-calling algorithms are used to produce the input short-reads. The length of the PhiX174 genome is 5386 bp, and Velvet [21] was used to perform the assembly. N50 is a statistic commonly used to assess the quality of de novo assembly. It is computed by sorting all contigs by their size in decreasing order and adding the length of these contigs until the sum is greater than 50% of the total length of all contigs. The length of the last added contig is reported as N50. A larger N50 indicates a better assembly. Also shown are the maximum contig length (denoted “Max”) and the total number of contigs (denoted “#Ctgs”).

Coverage	Bustard			Alta-Cyclic			BayesCall			naiveBayesCall		
	N50	Max	#Ctgs	N50	Max	#Ctgs	N50	Max	#Ctgs	N50	Max	#Ctgs
5X	145	153	277	140	146	251	146	156	358	146	158	349
10X	203	368	2315	200	353	2148	203	368	2435	203	365	2467
15X	352	685	4119	331	637	4047	368	712	4249	371	716	4263
20X	675	1162	4941	674	1119	4893	752	1246	5004	750	1259	5015

in base-call accuracy delivered by our algorithm implies that it is possible to obtain longer reads for a given error tolerance.

In our method, it is feasible to estimate parameters using a training set consisting of only a few hundred randomly chosen clusters. We believe that naiveBayesCall’s ability to estimate local parameters using a small number of clusters should allow one to take into account the important differences between different tiles and lanes (recall the results discussed in Section 4.3). One possible approach to take in the future is to partition the lane into several regions and estimate region-specific parameters. Further, adopting the following strategy may work well: Estimate lane-wide parameters using a small number of clusters randomly chosen from the entire lane. Then, obtain tile-specific or region-specific parameter estimates by initializing with the lane-wide estimates and by performing a few iterations of the expectation-maximization algorithm (as described in [9]) using a small number of clusters from the tile or region. We believe that the accuracy of naiveBayesCall can be improved significantly by using tile-specific or region-specific parameter estimates.

## Acknowledgments

We thank Kristian Stevens for useful discussion. This research is supported in part by an NSF CAREER Grant DBI-0846015, an Alfred P. Sloan Research Fellowship, and a Packard Fellowship for Science and Engineering.

## References

1. Bentley, D.R.: Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.* 16, 545–552 (2006)
2. Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W.L., Russ, C., Lander, E.S., Nusbaum, C., Jaffe, D.B.: Quality scores and SNP detection in sequencing-by-synthesis systems. *Genome Res.* 18, 763–770 (2008)



3. Butler, J., MacCallum, I., Kleber, M., Shlyakhter, I.A., Belmonte, M.K., Lander, E.S., Nusbaum, C., Jaffe, D.B.: ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research* 18(5), 810–820 (2008)
4. Chaisson, M.J.P., Brinza, D., Pevzner, P.A.: De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome research* (2008)
5. Erlich, Y., Mitra, P., Delabastide, M., McCombie, W., Hannon, G.: Alta-Cyclic: a self-optimizing base caller for next-generation sequencing. *Nat. Methods* 5, 679–682 (2008)
6. Ewing, B., Green, P.: Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Research* 8(3), 186–194 (1998)
7. Hellmann, I., Mang, Y., Gu, Z., Li, P., Vega, F.M.D.L., Clark, A.G., Nielsen, R.: Population genetic analysis of shotgun assemblies of genomic sequences from multiple individuals. *Genome Res.* 18(7), 1020–1029 (2008)
8. Jiang, R., Tavaré, S., Marjoram, P.: Population genetic inference from resequencing data. *Genetics* 181(1), 187–197 (2009)
9. Kao, W.C., Stevens, K., Song, Y.S.: BayesCall: A model-based basecalling algorithm for high-throughput short-read sequencing. *Genome Research* 19, 1884–1895 (2009)
10. Kiefer, J.: Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society* 4, 502–506 (1953)
11. Langmead, B., Trapnell, C., Pop, M., Salzberg, S.: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 25, R25 (2009)
12. Li, H., Ruan, J., Durbin, R.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* 18, 1851–1858 (2008)
13. Li, L., Speed, T.: An estimate of the crosstalk matrix in four-dye fluorescence-based DNA sequencing. *Electrophoresis* 20, 1433–1442 (1999)
14. Medvedev, P., Brudno, M.: Ab Initio Whole Genome Shotgun Assembly with Mated Short Reads. In: Vingron, M., Wong, L. (eds.) *RECOMB 2008*. LNCS (LNBI), vol. 4955, pp. 50–64. Springer, Heidelberg (2008)
15. Metzker, M.L.: Emerging technologies in DNA sequencing. *Genome Res.* 15(12), 1767–1776 (2005)
16. Rougemont, J., Amzallag, A., Iseli, C., Farinelli, L., Xenarios, I., Naef, F.: Probabilistic base calling of Solexa sequencing data. *BMC Bioinformatics* 9, 431 (2008)
17. Sundquist, A., Ronaghi, M., Tang, H., Pevzner, P., Batzoglu, S.: Whole-genome sequencing and assembly with high-throughput, short-read technologies. *PLoS One* 2(5), e484 (2007)
18. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13(2), 260–269 (1967)
19. Whiteford, N., Skelly, T., Curtis, C., Ritchie, M., Lohr, A., Zaranek, A., Abnizova, I., Brown, C.: Swift: Primary Data Analysis for the Illumina Solexa Sequencing Platform. *Bioinformatics* 25(17), 2194–2199 (2009)
20. Yin, Z., Severin, J., Giddings, M.C., Huang, W.A., Westphall, M.S., Smith, L.M.: Automatic matrix determination in four dye fluorescence-based DNA sequencing. *Electrophoresis* 17, 1143–1150 (1996)
21. Zerbino, D.R., Birney, E.: Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18(5), 821–829 (2008)

# Extracting Between-Pathway Models from E-MAP Interactions Using Expected Graph Compression

David R. Kelley and Carl Kingsford\*

Center for Bioinformatics and Computational Biology,  
Institute for Advanced Computer Studies, and Department of Computer Science,  
University of Maryland, College Park  
`carlk@cs.umd.edu`

**Abstract.** Genetic interactions (such as synthetic lethal interactions) have become quantifiable on a large-scale using the epistatic miniarray profile (E-MAP) method. An E-MAP allows the construction of a large, weighted network of both aggravating and alleviating genetic interactions between genes. By clustering genes into modules and establishing relationships between those modules, we can discover compensatory pathways. We introduce a general framework for applying greedy clustering heuristics to probabilistic graphs. We use this framework to apply a graph clustering method called graph summarization to an E-MAP that targets yeast chromosome biology. This results in a new method for clustering E-MAP data that we call Expected Graph Compression (EGC). We validate modules and compensatory pathways using enriched Gene Ontology annotations and a novel method based on correlated gene expression. EGC finds a number of modules that are not found by any previous methods to cluster E-MAP data. EGC also uncovers core sub-modules contained within several previously found modules, suggesting that EGC can reveal the finer structure of E-MAP networks.

## 1 Introduction

A synthetic sickness or lethality (SSL) interaction specifies a genetic dependence between two nonessential genes in which a double-knockout mutant has significantly reduced fitness. A number of studies have searched for such interactions and generated novel hypotheses about the cellular functions of genes of interest and the genetic network of the organism as a whole [24,21]. Two genes with an SSL interaction are believed to have a compensatory relationship: when one gene is lost, the other compensates for the lost gene's cellular function, whether directly or indirectly [10].

Following initial studies of SSL, Collins et al. pioneered the epistatic miniarray profile (E-MAP) method for defining genetic interactions quantitatively using data generated by the synthetic genetic array protocol [23,9]. An E-MAP aims to

---

\* Corresponding author.

generate a full matrix of interactions for a set of genes (e.g. 743 genes in [9]). For every pair of genes, the E-MAP reports the difference between the measured fitness of a double-knockout mutant strain and the expected fitness of the double knockout, computed using measured fitnesses of single-knockout mutant strains. One strength of the E-MAP approach is that the reported value can be used to measure the intensity of the interaction. An SSL interaction would result in a negative value, called an aggravating interaction, because the actual fitness would be much less than expected. Significantly positive values in the E-MAP, called alleviating interactions, suggest that the measured fitness is greater than expected. Such an interaction can occur when the performance of a cellular function is dependent on the presence of both genes and the loss of one of the genes is enough to disrupt the function. Then the loss of the second gene does not cause as much harm as might be expected. The identification of both aggravating and alleviating interactions is another strength of the E-MAP approach.

An effective method for analyzing genetic interactions is to find clusters within a graph where genes are represented by vertices and interactions are represented by edges. A number of studies searched these graphs for structures called between-pathway models (BPMs), which consist of two clusters, or modules, with many aggravating edges between them [14,25,16,4]. The genes in these modules are likely to be functionally related, and the pair of modules in a BPM are likely to have a relationship where one can compensate for a breakdown of the other. The quintessential example of such a structure would be two redundant pathways working towards a common downstream function and each with the property that if one gene is knocked out, its pathway is disabled. In this case, we would expect to see many SSL edges for genes between the two pathways. Though things are seldom this simple in the *S. cerevisiae* genetic network, previous studies have found many convincing examples of BPMs. More recent work has performed a similar clustering analysis on interaction networks generated by E-MAP, taking into account both aggravating and alleviating interactions and the interaction intensities [26,2].

In general, the network clustering problem is to partition the nodes of a graph  $G$  into disjoint subsets  $X = \{M_1, \dots, M_k\}$  so that some measure  $f(G, X)$  of the quality of the partitioning is optimized. The methods of modularity [20], graph summarization (GS) [18], minimum multiway cut [27], and others all fall under this general framework. Many clustering quality functions are known to be NP-hard to optimize (e.g. [5]) or are conjectured to be NP-hard (such as GS). Often the best algorithms in practice for these difficult clustering problems are agglomerative hierarchical clustering approaches, e.g. [18,7]. In these approaches, each node is initially placed in its own module, and pairs of modules are successively merged to improve the clustering quality. We show how to extend greedy agglomerative clustering methods with certain locality properties of the quality function to probabilistic graphs, where each edge  $e$  exists with probability  $p(e)$ .

We apply this greedy clustering framework to probabilistic graphs derived from yeast E-MAP data to uncover clusters and BPMs. We use a clustering quality (cost) function derived from GS [18]. GS is an approach to graph partitioning

based on compression that is well-founded in information theory [18] and was successful in finding functionally cohesive modules in the *S. cerevisiae* protein-protein interaction network such that unannotated proteins could be accurately annotated based on the modules they were placed in [19]. A particular strength of GS is finding approximate bicliques in the graph, which are identified as compressible structures. Because approximate bicliques are the subgraph signature of a BPM, the GS framework is a good clustering formulation for finding BPMs in a genetic interaction network. The framework is sufficiently robust that it can incorporate domain-specific information such as physical interactions between proteins, as was found to be important in previous studies [26,12]. We call our approach expected graph compression (EGC).

Proper validation of hypothesized BPMs is important but challenging because there is no ground truth against which to compare. We validate BPMs using two new tests that employ *S. cerevisiae* gene expression measurements aggregated from 132 studies [12] based on intra- and inter-module expression correlation. BPMs uncovered by EGC and previous E-MAP clustering methods demonstrate greater correlation of gene expression than expected by chance. We also validate modules and BPMs by enrichment of Gene Ontology (GO) annotations [1]. The modules and BPMs found by EGC compare well to those from previous studies and cover the largest number of unique GO annotations. EGC returns more modules and BPMs than past methods at a similar level of quality. A substantial number of modules found are novel and dissimilar to any modules reported by prior studies. In addition, a subset of modules are uncovered by all existing algorithms, and we consider these to be super-validated. Finally, in a number of cases, two or more EGC modules are entirely contained within a module found by a previous study, representing submodules of finer resolution.

In summary, we show how to derive an efficient greedy heuristic for optimizing the expected quality of a graph partitioning for a general class of quality measures. We apply this technique to E-MAP data to uncover modules and BPMs that are of comparable quality to existing E-MAP clustering approaches according to two novel validation measures and GO annotations. We also show that many of the modules found by this method better match known biological units and can reveal the fine structure of compensatory pathways.

## 2 Methods

### 2.1 Agglomerative Hierarchical Clustering in Probabilistic Graphs

A *probabilistic graph*  $\mathcal{G}$  is a triple  $(V_{\mathcal{G}}, E_{\mathcal{G}}, p_{\mathcal{G}})$ , where  $V_{\mathcal{G}}$  is the vertex set,  $E_{\mathcal{G}}$  is the edge set, and  $p_{\mathcal{G}} : E_{\mathcal{G}} \rightarrow [0, 1]$  is a function such that  $p_{\mathcal{G}}(e)$  is the probability that edge  $e \in E_{\mathcal{G}}$  exists, independent of all other edges in the graph. Pairs of vertices not connected by an edge in  $E_{\mathcal{G}}$  have probability 0 of an edge. Probabilistic graphs arise in several applications of clustering biological networks, and several schemes [15,17] have been proposed to assign probabilities to edges in physical interaction networks and functional association networks. A probabilistic graph  $\mathcal{G}$  may alternatively be viewed as the set of graphs obtained by

choosing a subset of edges that have non-zero probability. Under this view, any definite (non-probabilistic) graph  $g \in \mathcal{G}$  represents a single instantiation of  $\mathcal{G}$  where all edges have been determined to exist or not. We take this view for the remainder of this section.

The graph clustering problem on non-probabilistic graphs seeks to partition the nodes into subsets  $X = \{M_1, \dots, M_k\}$  to minimize some cost function  $f(\mathcal{G}, X)$ . When dealing with probabilistic graphs, instead of computing  $\operatorname{argmin}_X f(\mathcal{G}, X)$ , a more natural goal is to find

$$\operatorname{argmin}_X \mathbb{E}_{g \in \mathcal{G}} f(g, X). \tag{1}$$

In other words, we look for the partitioning  $X$  that minimizes the expected cost of the partitioning over all possible instantiations of the probabilistic graph  $\mathcal{G}$ . Minimizing (1) can be more difficult than the non-probabilistic setting because of the large number of possible graphs that must be considered.

If the clustering cost function  $f$  has the following two properties then heuristics for optimizing (1) exist that are only slightly less efficient than the heuristics for optimizing the non-probabilistic variant:

- (i)  $f$  can be decomposed into a sum of pairwise costs  $h$  between modules.

$$f(\mathcal{G}, X) = \sum_{M_i, M_j \in X} h(M_i, M_j, \mathcal{G}). \tag{2}$$

- (ii)  $h(M_i, M_j, \mathcal{G})$  depends only on  $A_{ij}$ , the number of edges between  $M_i$  and  $M_j$ .

Admittedly, properties (i) and (ii) are restrictive on possible cost functions  $f$ . However, GS satisfies both and so good heuristics for minimizing its cost function exist for probabilistic graphs, as we now show.

Suppose a clustering cost function  $f$  satisfies properties (i) and (ii) above. Then, by property (i), we can rewrite (1) as follows:

$$\mathbb{E}_{g \in \mathcal{G}} f(g, X) = \sum_{g \in \mathcal{G}} P(g) \sum_{M_i, M_j \in X} h(M_i, M_j, g) = \sum_{M_i, M_j \in X} \sum_{g \in \mathcal{G}} P(g) h(M_i, M_j, g). \tag{3}$$

Rather than sum over all  $g \in \mathcal{G}$  as in (3), by property (ii), we can collect together all graphs that have a particular value for  $A_{ij}$  between modules  $M_i$  and  $M_j$  and sum over possible values for  $A_{ij}$ :

$$\mathbb{E}_{g \in \mathcal{G}} f(g, X) = \sum_{M_i, M_j \in X} \sum_{a \in \operatorname{range}(A_{ij})} P(A_{ij} = a \mid \mathcal{G}) h(M_i, M_j, a). \tag{4}$$

Here  $P(A_{ij} = a \mid \mathcal{G})$  is the probability that a graph in  $\mathcal{G}$  has  $a$  edges between modules  $M_i$  and  $M_j$  and  $\operatorname{range}(A_{ij})$  is the set of possible values for  $A_{ij}$ .

To compute the expectation of the cost function efficiently without analyzing every graph, we store the probabilities  $P(A_{ij} = a \mid \mathcal{G})$  for  $a \in \operatorname{range}(A_{ij})$  for every pair of modules and update them each time two modules are merged together. At the start of the algorithm, each vertex is in its own module, and

$P(A_{ij} = 1 \mid \mathcal{G}) = p_{\mathcal{G}}(\{i, j\})$  and  $P(A_{ij} = 0 \mid \mathcal{G}) = 1 - p_{\mathcal{G}}(\{i, j\})$ , where  $p_{\mathcal{G}}(\{i, j\})$  is the probability that edge  $\{i, j\}$  exists. When modules  $M_i$  and  $M_j$  are merged to form a new module  $M_k$ , the probabilities for the number of edges between  $M_k$  and any other module  $M_q$  are calculated by

$$P(A_{kq} = a) = \sum_{x=0}^a P(A_{iq} = x)P(A_{jq} = a - x). \tag{5}$$

Considered as a vector of probabilities,  $P(A_{kq}) = \langle P(A_{kq} = 0), P(A_{kq} = 1), \dots \rangle$  can be computed as the convolution of the vectors  $P(A_{iq})$  and  $P(A_{jq})$ . Using the fast Fourier transform algorithm, the new probabilities can be computed in time  $O(\pi_{ij} \log \pi_{ij})$ , where  $\pi_{ij}$  is the number of possible edges between modules  $i$  and  $j$ . Thus, we can compute the expected cost of an edge at any stage of the greedy algorithm with little extra work.

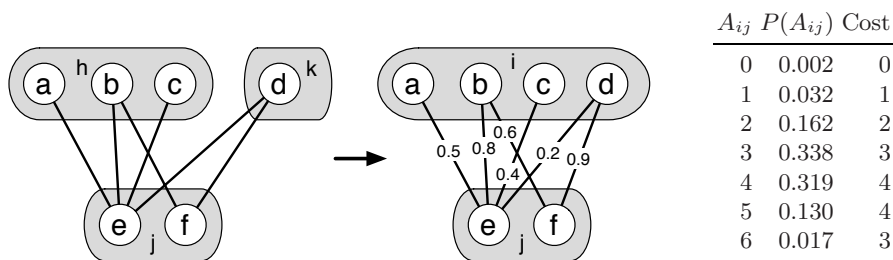
## 2.2 Expected Graph Compression

Graph summarization (GS) is a graph compression algorithm that has been successfully used to cluster protein-protein interaction networks [19]. GS seeks to minimize the cost to represent all edges in a (non-probabilistic) graph  $G = (V_G, E_G)$  by searching for a new graph  $H = (V_H, E_H)$  where a node in  $V_H$  represents one or more nodes in  $V_G$ . A summarizing edge  $(A, B)$  in  $H$  implies that there is a biclique in  $G$  between all nodes that comprise nodes  $A$  and  $B$  and serves to represent those biclique edges in  $E_G$  more compactly. In the common case that the nodes contained in  $A$  and  $B$  have many edges between them but do not form a complete bipartite graph, we may still use a summarizing edge and add *corrections* to remove the missing edges. Edges in  $E_G$  that are not represented by any summarizing edges in  $E_H$  also must be added as corrections. The new graph  $H$  and list of removal corrections RC and addition corrections AC can be used to completely reconstruct  $G$ . The cost of this partitioning is given by  $f_{\text{gs}}(G, H) = |E_H| + |\text{RC}| + |\text{AC}|$ . The most effective algorithm in previous studies to minimize  $f_{\text{gs}}(G, H)$  was a greedy algorithm that at every iteration merges the two nodes of  $H$  that would most reduce the graph cost  $f_{\text{gs}}$  [18, 19] (see Figure 1). Then each vertex of  $H$  induces a module of vertices in  $G$ .

The framework described above can be used to extend GS to probabilistic graphs. We first note that  $f_{\text{gs}}$  obeys property (i) and can be rewritten as a sum of pairwise costs as in (2) where  $h_{\text{gs}}$  gives the cost to represent the edges between two modules  $M_i, M_j$ . This cost depends on whether we can represent the edges from the original graph between these modules less expensively with a summarizing edge in  $E_H$  or by listing each of the original edges individually as addition corrections. Specifically, we have

$$h_{\text{gs}}(M_i, M_j, \mathcal{G}) = h_{\text{gs}}(M_i, M_j, A_{ij}) = \min\{\pi_{ij} - A_{ij} + 1, A_{ij}\}, \tag{6}$$

where  $\pi_{ij}$  is the number of possible edges between modules  $M_i$  and  $M_j$ , which is  $|M_i||M_j|$  when  $i \neq j$ , and  $A_{ij}$  is the number of edges from the original graph (or



**Fig. 1.** Every iteration of the EGC greedy optimization algorithm merges the two modules, such as  $h$  and  $k$  to  $j$  above, that result in the largest decrease in expected cost as given by (4). The table on the right outlines the necessary information for this calculation, including the probability of  $A_{ij} = a$  edges between modules  $i$  and  $j$  and the cost of  $A_{ij} = a$  edges between the modules defined by (6).

a single instantiation in the probabilistic case) where one vertex belongs to  $M_i$  and the other belongs to  $M_j$ . Thus, the cost of an edge depends on the graph only through  $A_{ij}$  and  $f_{gs}$  obeys property (ii). Therefore, the general framework described above can be used to find a compressed graph  $H$  that has the minimum expected cost.

Due to the stochasticity in the graph, this expected graph compression (EGC) abandons the premise of a lossless compression upon which GS was built. In return, the edge weights allow greater discrimination in determination of modules. If the edge probabilities are all 1, EGC outputs the same summarizing graph  $H$  as GS. As the probabilities decrease to 0, the modules gradually break apart, as there is less to gain in trying to compactly represent improbable edges. Using the edge weights allows us to emphasize the summarization of interactions with high weight while still taking interactions with low weight into account.

### 2.3 Application to E-MAP Data

The yeast chromosome E-MAP data [9] forms a  $743 \times 743$  matrix, containing 183040 interaction values (some interaction experiments fail) that can take on any real number ( $> 0$  indicates alleviating and  $< 0$  indicates aggravating interactions). An effective way to make use of the values is to build a model (e.g. a mixture of Gaussian distributions representing different interaction classes [26]). Our approach is to map each value to a probability that the interaction truly exists. This model accounts for experimental noise and ensures that the probability of interaction increases with increasing magnitude of the E-MAP value. An interaction value  $< -3$  has been used previously as a cutoff for an SSL interaction [13], implicitly assigning a probability of 1 to interactions below the cutoff and 0 to those above it. Using a logistic function that softens this threshold, we map an E-MAP value  $x$  to probability  $(1 + e^{-3|x|+7.5})^{-1}$ , which has probability 1/2 at E-MAP value 2.5 where it rises most steeply. An E-MAP value of 3 gives a probability of  $\sim 0.82$  of interaction. The resulting probabilistic graph contained



20763 aggravating edges and 10581 alleviating edges representing interactions that mapped to probabilities  $> 0.05$ . The results were robust to minor parameter changes.

Alleviating and aggravating interactions indicate a very different type of relationship between two genes. We separate the two types of edges into different probabilistic graphs,  $\mathcal{G}_{ag}$  for aggravating and  $\mathcal{G}_{al}$  for alleviating, and search for the partitioning  $X$  that minimizes the expected cost to represent both graphs:

$$\operatorname{argmin}_X \mathbb{E}_{g \in \mathcal{G}_{ag}} f_{gs}(g, X) + \mathbb{E}_{g \in \mathcal{G}_{al}} f_{gs}(g, X). \quad (7)$$

Because the vertex set is the same in both  $\mathcal{G}_{ag}$  and  $\mathcal{G}_{al}$ , we can simplify (7) so that both types of interactions are considered simultaneously when computing edge costs. The algorithm must also be adjusted to account for the high rate of failure for E-MAP experiments ( $\sim 1/3$  of the matrix in the data set used [9]). In order to not overestimate the number of possible edges  $\pi_{ij}$  between two modules, we exclude untested or failed pairs of proteins from  $\pi_{ij}$ .

Previous work has demonstrated that compensatory pathways are more readily identified when physical interactions are simultaneously considered, specifically between proteins within modules [14,26,2]. To make use of physical interactions within EGC we only consider a merge between two modules if there is a physical interaction between a pair of proteins with one protein in each module. This way EGC will only output connected components of proteins as modules. EGC was run using two different sets of physical interactions in order to compare to previous methods. One set includes 2061 physical interactions used by Ulitsky et al. [26]. The interactions were originally downloaded from the SGD and BioGrid databases [6,22] and exclude those found using the two-hybrid method. Most analysis in this paper focuses on the EGC results with this interaction set. The second set was used by Bandyopadhyay et al. [2] and combines interactions from two large-scale TAP-MS studies [8]. In this set, real values measure the evidence for each interaction. We used the 1552 interactions with scores  $> 1$ , a threshold used for analysis by Bandyopadhyay et al.

Another desirable attribute of a clustering algorithm is the ability to guide the approximate size of the final modules. As described, EGC finds modules with a small average size and many modules of size 2 and 3. Although small modules are interesting, slightly larger ones would be more biologically informative. EGC offers a straightforward way to encourage the algorithm to grow the modules larger. If the cost of a removal correction is lowered, there is more incentive to combine genes into modules as the missing interactions are less costly. We computed the validation metrics for removal correction costs of 1.0, 0.5, 0.25, and 0.1 and found 0.25 to be the most effective in creating reasonably sized modules that validated well.

Past studies on GS [18,19] actually used a variation of the greedy optimization algorithm where the reductions in cost used to evaluate possible merges were normalized by the sum of all edge costs for the two modules to be merged. As originally formulated, the greedy algorithm tended to grow a small number of modules very large in the initial iterations to the detriment of the final global



cost. Normalizing the cost reduction encourages a more balanced progression. We implement this variation in EGC.

After running EGC, we defined modules as the sets of genes merged into vertices in  $V_H$  that contain more than one gene. If the probability of a summarizing edge is  $> 0$  between two modules, we consider the pair to be a BPM.

## 2.4 Validation via GO Annotations

True authentication of a BPM would require direct lab demonstration of one set of proteins compensating for the other. Since doing so in a large-scale fashion would be infeasible, we resort to indirect computational tests. For example, if the proteins in one module truly compensate for proteins in the other module, we might expect the modules to perform related functions. This may not be the case for all BPMs, but two functionally related modules should be better candidates for a compensatory relationship than two functionally disjoint modules.

One way to assess functional similarity is to compare Gene Ontology (GO) [11] annotations. We associate GO terms with modules using the FuncAssociate web service [3], which searches all 3 subontologies (biological process, cellular component, and molecular function) of the Gene Ontology. FuncAssociate performs a hypergeometric enrichment test and corrects for multiple hypotheses using simulation. We consider a module to be validated if it is enriched for a GO annotation with P-value  $< 0.05$ . We additionally require the annotation to apply to  $\leq 500$  *S. cerevisiae* proteins as terms that describe  $> 500$  proteins were too vague.

We introduce a similar test for BPMs by comparing the enriched annotations for each module in the BPM. For each pair of enriched annotations (one for each module), we find their lowest common ancestor in the GO hierarchy. If it applies to  $\leq 500$  *S. cerevisiae* proteins, we consider the module annotations to be functionally related and the BPM to be validated.

## 2.5 Validation via Gene Expression

Hescott et al. recently introduced a promising validation method for BPMs that uses gene expression measurements from mutant strains of *S. cerevisiae* with a single gene knocked out [11]. However, this approach is hindered by the limited number of such expression data sets. On the other hand, gene expression data for healthy cells in a wide range of conditions is plentiful. Correlated gene expression for proteins in a module across various cellular stages and conditions is evidence of functional coherence. If gene expression is correlated between modules in a BPM, they are likely to be functionally related, thus supporting the BPM.

We use an aggregated set of experiments from 132 yeast gene expression studies [12]. To test a module, we first compute the Pearson correlation coefficient between expression vectors for every pair of genes in the module. We use the average of these correlations to measure the coherence of gene expression for the module. To assess the significance of the statistic for a module of size  $N$ , we computed it for 1000 randomly sampled sets of  $N$  genes from the E-MAP data set. We report a P-value for each module that is equal to the proportion of

**Table 1.** Comparison of modules output by variations of EGC and previous methods

	Modules	GO Annotations <sup>a</sup>	Annotated <sup>b</sup>	Correlated <sup>c</sup>
EGC <sup>1</sup>	118	636	89.8% (106)	23.7% (28)
EGC B-phys <sup>2</sup>	113	629	85.8% (97)	23.0% (26)
EGC no al <sup>3</sup>	128	641	85.9% (110)	20.3% (26)
EGC r=1 <sup>4</sup>	129	641	85.3% (110)	19.4% (25)
EGC no phys <sup>5</sup>	243	607	46.5% (113)	12.85% (31)
Ulitsky et al.	62	588	100% (62)	32.3% (20)
Bandyopadhyay et al.	91	624	85.5% (76)	24.2% (22)

<sup>1</sup>considers physical interactions (from the Ulitsky et al. data set) and alleviating interactions and sets the removal correction cost to 0.25; <sup>2</sup>uses the Bandyopadhyay et al. physical interactions; <sup>3</sup>uses no alleviating interactions; <sup>4</sup>removal correction cost = 1; <sup>5</sup>no physical interactions considered. <sup>a</sup>number of unique GO terms that are enriched in some module; <sup>b</sup>number of modules enriched with at least 1 GO term; <sup>c</sup>modules that attain an expression correlation with P-value < 0.05.

randomly sampled gene sets that had a greater correlation statistic. We consider modules with a P-value < 0.05 to be validated under this test.

To test BPMs, we create a centroid expression vector for each module where entry  $i$  in the vector is the average of the expression values for the module's genes in experiment  $i$ . For pairs of modules that participate in a BPM, we compute the correlation between their centroid expression vectors over all experiments where  $\geq 2$  genes were measured in each module. We again survey the background distribution by sampling pairs of sets of randomly chosen genes with sizes matching the modules in the BPMs being tested and computing the centroid correlation statistic. BPMs with P-values < 0.05 were considered validated.

### 3 Results and Discussion

#### 3.1 Modules Uncovered by Expected Graph Compression

EGC was run on the yeast chromosome E-MAP generated by Collins et al. [9]. Descriptions of all modules and BPMs can be found at <http://www.cbcb.umd.edu/research/bionet/EGC>. EGC has a single parameter,  $r$ , which is the cost of a removal correction. Standard GS uses  $r = 1$ , but setting  $r = 0.25$  produced larger (3.0 genes per module with  $r = 2.5$  versus 2.3 with  $r = 1$ ) and more biologically relevant modules. This leads to greater sensitivity and precision of GO term annotation, seen in rows "EGC" and "EGC r=1" in Tables 1 and 2. With respect to correlation of gene expression, a greater percentage of modules and BPMs are validated when using  $r = 0.25$  compared with  $r = 1$ .

While SSL (aggravating) interactions are widely used for BPM identification [14, 25, 16, 4], E-MAPs describe both alleviating and aggravating interactions. To assess the significance of alleviating interactions on BPM quality, we compare modes of EGC with and without them. If only aggravating interactions are

**Table 2.** Comparison of BPMs output by variations of EGC and previous methods

	BPMs	GO Annotations <sup>a</sup>	Annotated <sup>b</sup>	Correlated <sup>c</sup>
EGC	403	175	43.4% (175)	12.9% (52)
EGC B-phys	369	165	49.3% (182)	13.0% (48)
EGC no al	371	175	44.7% (166)	12.9% (48)
EGC r=1	629	190	43.9% (276)	10.0% (63)
EGC no phys	706	159	28.5% (201)	9.2% (65)
Ulitsky et al.	153	141	66.0% (101)	15.7% (24)
Bandyopadhyay et al.	208	132	45.2% (94)	13.0% (27)

<sup>a</sup>number of unique GO terms that satisfy the criteria in Section 2.4 for at least 1 BPM;

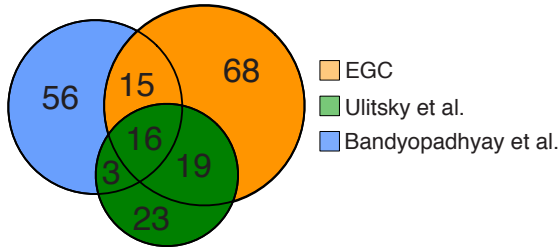
<sup>b</sup>percent of BPMs annotated with at least one term. <sup>c</sup>percent of BPMs for which the centroid correlation test described in Section 2.5 attains a P-value < 0.05.

considered in EGC, we find fewer BPMs and modules that are less likely to be enriched for GO annotations and less correlated in gene expression (“EGC no al” in Tables 1 and 2). Though the impact of alleviating interactions is not overwhelming, the results suggest that they should be included. Overall, EGC places 3059 aggravating and 492 alleviating interactions across modules in BPMs and 81 aggravating and 159 alleviating interactions within modules. Alleviating interactions will appear within modules when the loss of one gene disrupts the module’s function in such a way that the loss of more genes does not have a large affect on fitness.

Physical interactions were demonstrated to be helpful in previous efforts to cluster E-MAP networks [26,2], but not all successful studies on BPMs have incorporated them [4]. We find that using physical interactions to restrict the greedy merges leads to a major boost in performance shown in rows “EGC” and “EGC no phys” in Tables 1 and 2. In the EGC greedy algorithm, there are commonly a number of conflicting candidates with similar profiles of genetic interactions, and the physical interactions serve as a second independent source of evidence that the proteins being considered are functionally related. The set of physical interactions used by Bandyopadhyay et al. gave results of a similar quality — the modules are slightly worse and the BPMs are slightly better.

### 3.2 Comparison with Previous Studies

We compare the EGC modules and BPMs with those from previous studies performed by Ulitsky et al. [26] and Bandyopadhyay et al. [2] on the yeast chromosome E-MAP. All modules and BPMs are considered in these comparisons because the validation tests are inexact and even modules and BPMs that are not validated represent sets of genes with potentially interesting genetic interaction patterns. EGC produces many more modules than either of the previous methods (Table 1). These modules cover slightly more GO annotations (636 vs. 588 and 624), and a higher percentage of these modules are enriched for an annotation than Bandyopadhyay et al. A higher percentage of Ulitsky et al. modules are



**Fig. 2.** Modules from the 3 methods publishing results on the yeast chromosome E-MAP [9] are displayed as a Venn diagram. Two modules are considered equivalent if their Jaccard index is  $\geq 2/3$  (which does not account for containment of one module in another). Each study contributes unique modules, with EGC offering the most.

annotated and have correlated gene expression, though that algorithm produced far fewer modules. Hence, EGC produces many more modules of comparable or slightly lower quality than existing E-MAP clustering approaches.

EGC also naturally identifies modules that are dense with interactions between genes within the module. These modules are tagged by EGC by creating self-edges that summarize within-module interactions. EGC identifies 38 of these dense modules. They are generally high quality as all 38 are enriched for a GO annotation and 12 (31.6%) have correlated gene expression.

We also tested the algorithm of Brady et al. [4] who reported compelling results on a set of unweighted SSL interactions. We provided this method with E-MAP interactions that had value  $< -3$  as input because these most closely represent the SSL edges for which it was designed. However, the algorithm was less successful when taken out of its intended context in this way. The modules it returned were annotated with fewer GO terms (571) at a lower precision (75% modules annotated) and a lower percentage were validated by correlation of gene expression (15%) compared to the methods designed specifically for E-MAPs.

EGC tends to find smaller modules than the other methods — the mean module size is 3.0 for EGC, 5.0 for Ulitsky et al., and 4.1 for Bandyopadhyay et al. — including many modules of size 2. These are pairs of genes that have very similar profiles of genetic interactions, and no other genes were sufficiently similar to have been merged into the cluster. Though EGC does not grow the modules as large as the other algorithms, it places more genes in modules than Ulitsky et al. (355 vs. 313). (In each algorithm, many genes exist as singletons that are not placed in any module.) Bandyopadhyay et al. placed slightly more genes into modules (374). More BPMs are identified per module by EGC (average 6.8) than other methods (4.9 for Ulitsky et al. and 4.6 for Bandyopadhyay et al.).

### 3.3 Novel and Super-Validated Modules

Many modules are found only by EGC. We consider two modules to be equivalent if the Jaccard index between them is  $\geq 2/3$ . The Jaccard index between two

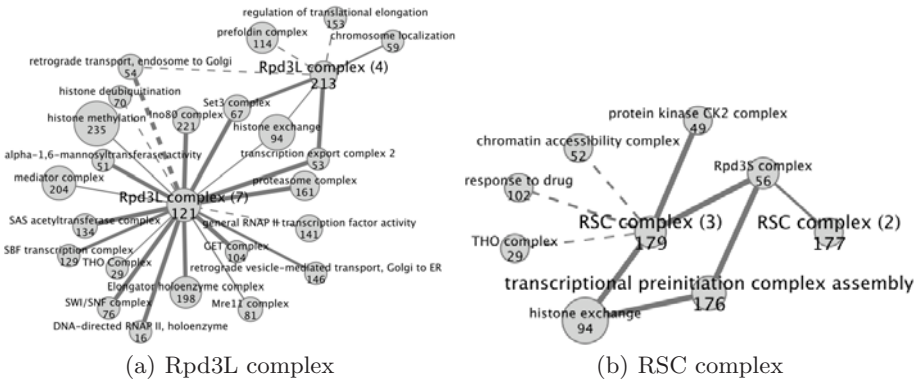
modules is the number of proteins in both divided by the number of proteins in at least one. Figure 2 shows a Venn diagram of the module overlaps using this equivalence, which demonstrates the substantial unique contributions of each method. If we further exclude modules for which  $> 90\%$  of their proteins are contained in a module from another study, we obtain a collection of modules that represent groupings of genes that are truly unique to each study. EGC produces 37 unique modules (31%), 27 of which are enriched for a GO annotation and 4 of which display correlation of gene expression. These 37 modules participate in 128 BPMs, 44 of which are validated by the annotation test and 7 of which display correlation of gene expression. By the same criteria, 23 (37%) of the modules from Ulitsky et al. and 51 (56%) of the modules from Bandyopadhyay et al. are unique. The large fraction of unique modules produced by each method indicates that the underlying motivations for each approach are uncovering complementary views of the E-MAP network and suggests that there is some uncertainty about what the “true” clustering should be.

In one example of a unique module, EGC finds a module of 12 genes (containing e.g. CDC73) that is both annotated and displays correlation of gene expression. This module contains all 4 yeast genes annotated as histone ubiquitination, which other methods do not place together, and 11 genes annotated as histone methylation. The module participates in 14 BPMs, including aggravating edges to modules annotated as histone exchange; GET complex; and retrograde transport, endosome to Golgi. Another unique module has 4 genes (RAD9, RAD24, DUN1, RAD53), 3 of which are involved in a DNA damage signal transduction pathway. This module is involved in 7 BPMs, such as with modules annotated as replication fork protection, Ctf18 RFC-like complex, and DNA replication factor A complex.

Furthermore, 16 modules are found by every study. All 16 of these modules are annotated with GO terms and 8 have correlated gene expression, which suggests that they are highly reliable and should be considered the “super-validated” modules of the yeast chromosome E-MAP. For example, one module consists of all 6 proteins of the elongator holoenzyme complex (e.g. ELP6) and another consists of 6 of 7 proteins of the prefoldin complex (e.g. GIM3).

### 3.4 Using EGC to Find Submodules of Larger Modules

Because EGC produces smaller modules in general, it reveals the finer structure of the chromosome E-MAP network. We examined cases where EGC split a module from a previous study into smaller submodules. There exists 6 cases where  $\geq 2$  EGC modules are contained in a single Ulitsky et al. module and 7 cases where  $\geq 2$  EGC modules are contained in a single Bandyopadhyay et al. module. In most cases, genetic interactions and GO annotations support the split. For example, Ulitsky et al. report a module of 13 proteins, 11 of which are annotated as belonging to the Rpd3L complex. The 2 proteins (SPT3, SPT8) without this annotation are placed by EGC into a separate module, and the remaining 11 are divided into modules of size 4 (UME6, RXT3, UME1, CTI6)



**Fig. 3.** Nodes indicate EGC modules labeled by a module ID and annotation. Node size is proportional to the number of genes in the module. Edges indicate BPMs with solid lines as aggravating interactions and dashed lines as alleviating interactions. Edge thickness is proportional to the probability that a summarizing edge is used between the two modules (see 2.2). (a) EGC separates the Rpd3L complex found by Ulitsky et al. into two submodules based on vastly different genetic interaction profiles. (b) EGC divides another Ulitsky et al. module into three sets. Four genes annotated as transcriptional preinitiation complex assembly form a module, and five RSC complex genes are split into two modules, each with its own distinct pattern of genetic interactions.

and 7 (PHO23, RXT2, DEP1, SAP30, SDS3, RPD3, SIN3). While both submodules share a number of interactions (Figure 3(a)), each module also has unique interactions, including edges from the module of size 7 to modules annotated as DNA-directed RNA polymerase II holoenzxxyme, GET complex, and proteasome complex. Genetic interactions clearly dichotomize the Rpd3L complex, and EGC uncovers these submodules.

EGC breaks up another Ulitsky et al. module with 11 genes (Figure 3(b)). One submodule (TAF4, TAF12, TAF6, and TAF9) contains the only genes in the set annotated with transcription initiation factor activity. All genes in the submodule are annotated with transcriptional preinitiation complex assembly. Though five genes in the Ulitsky et al. module are annotated as part of the RSC complex, EGC splits them into two modules (RSC1, RSC8, RSC6 and RSC58, RSC9) based on very different genetic interaction profiles. While both modules have an aggravating edge to a module annotated as Rpd3S complex, the module with 3 genes also interacts with modules annotated as histone exchange, protein kinase CK2 complex, THO complex and more. Again, EGC is able to detect potential submodules in this protein complex supported by the E-MAP.

### 4 Conclusion

For many clustering tasks, a greedy strategy is a useful heuristic. Additionally, many types of networks can be modeled as weighted or probabilistic graphs.

We have outlined a general framework to perform greedy merging for clustering methods on probabilistic graphs. We applied this framework to identify modules and between-pathway models (BPMs) in a genetic interaction network. We introduced a pair of new validation tests based on correlation of gene expression, which are moderately successful in that the modules and BPMs are more likely to have correlated gene expression than expected by chance. However, the effect is not overwhelming and alternative relationships between the pairs of modules in BPMs cannot be ruled out. In many cases, two or more EGC modules are contained in a single module found by a previous method. We found that in most cases, GO annotations and genetic interactions support the division uncovered by EGC. These finer-grained modules may be useful to understand the structure of the larger modules found by previous methods.

## Acknowledgments

The authors thank Ben Langmead and Saket Navlakha for helpful discussions and comments on the manuscript. C.K. thanks the National Science Foundation for grants 0849899 and 0812111.

## References

1. Ashburner, M., Ball, C., Blake, J., et al.: Gene Ontology: tool for the unification of biology. *Nature Genetics* 25(1), 25–29 (2000)
2. Bandyopadhyay, S., Kelley, R., Krogan, N.J., Ideker, T.: Functional maps of protein complexes from quantitative genetic interaction data. *PLoS Comput. Biol.* 4(4), e1000065 (2008)
3. Berriz, G., King, O., Bryant, B., Sander, C., Roth, F.: Characterizing gene sets with FuncAssociate. *Bioinformatics* 19(18), 2502–2504 (2003)
4. Brady, A., Maxwell, K., Daniels, N., Cowen, L.J.: Fault tolerance in protein interaction networks: Stable bipartite subgraphs and redundant pathways. *PLoS ONE* 4(4), e5364 (2009)
5. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hofer, M., Nikoloski, Z., Wagner, D.: On finding graph clusterings with maximum modularity. In: Brandstädt, A., Kratsch, D., Müller, H. (eds.) *WG 2007*. LNCS, vol. 4769, pp. 121–132. Springer, Heidelberg (2007)
6. Cherry, J., Adler, C., Ball, C., et al.: SGD: *Saccharomyces* genome database. *Nucleic Acids Research* 26(1), 73 (1998)
7. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* 70(6), 66111 (2004)
8. Collins, S., Kemmeren, P., Zhao, X., et al.: Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol. Cell. Proteomics* 6(3), 439 (2007)
9. Collins, S.R., Miller, K.M., Maas, N.L., et al.: Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature* 446(7137), 806–810 (2007)
10. Hartman, J.L., Garvik, B., Hartwell, L.: Principles for the buffering of genetic variation. *Science* 291(5506), 1001–1004 (2001)



11. Hescott, B.J., Leiserson, M.D.M., Cowen, L.J., Slonim, D.K.: Evaluating between-pathway models with expression data. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 372–385. Springer, Heidelberg (2009)
12. Hibbs, M.A., Hess, D.C., Myers, C.L., Huttenhower, C., Li, K., Troyanskaya, O.G.: Exploring the functional landscape of gene expression: directed search of large microarray compendia. *Bioinformatics* 23(20), 2692–2699 (2007)
13. Ihmels, J., Collins, S.R., Schuldiner, M., Krogan, N.J., Weissman, J.S.: Backup without redundancy: genetic interactions reveal the cost of duplicate gene loss. *Mol. Syst. Biol.* 3 (2007)
14. Kelley, R., Ideker, T.: Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology* 23(5), 561–566 (2005)
15. Leach, S., Gabow, A., Hunter, L., Goldberg, D.S.: Assessing and combining reliability of protein interaction sources. In: Pac. Symp. Biocomput., pp. 433–444 (2007)
16. Ma, X., Tarone, A.M., Li, W.: Mapping genetically compensatory pathways from synthetic lethal interactions in yeast. *PLoS ONE* 3(4) (2008)
17. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21(suppl. 1) (June 2005)
18. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data, pp. 419–432. ACM, New York (2008)
19. Navlakha, S., Schatz, M.C., Kingsford, C.: Revealing biological modules via graph summarization. *Journal of Computational Biology* 16(2), 253–264 (2009)
20. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23), 8577–8582 (2006)
21. Pan, X., Yuan, D.S., Xiang, D., Wang, X., Sookhai-Mahadeo, S., Bader, J.S., Hiter, P., Spencer, F., Boeke, J.D.: A robust toolkit for functional profiling of the yeast genome. *Mol. Cell* 16(3), 487–496 (2004)
22. Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. *Nucleic Acids Res* 34(Database Issue), D535 (2006)
23. Tong, A.H., Evangelista, M., Parsons, A.B., et al.: Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* 294(5550), 2364–2368 (2001)
24. Tong, A.H., Lesage, G., Bader, G.D., et al.: Global mapping of the yeast genetic interaction network. *Science* 303(5659), 808–813 (2004)
25. Ulitsky, I., Shamir, R.: Pathway redundancy and protein essentiality revealed in the *Saccharomyces cerevisiae* interaction networks. *Mol. Syst. Biol.* 3 (April 2007)
26. Ulitsky, I., Shlomi, T., Kupiec, M., Shamir, R.: E-MAPs to module maps: dissecting quantitative genetic interactions using physical interactions. *Mol. Syst. Biol.* 4 (July 2008)
27. Vazquez, A., Flammini, A., Maritan, A., Vespignani, A.: Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology* 21(6), 697–700 (2003)



# Simultaneous Identification of Causal Genes and Dys-Regulated Pathways in Complex Diseases

Yoo-Ah Kim, Stefan Wuchty, and Teresa M. Przytycka

National Center for Biotechnology Information, NLM, NIH, 8600 Rockville Pike,  
Building 38A, Bethesda, MD 20894  
{kimy3,wuchtys,przytyck}@ncbi.nlm.nih.gov

**Abstract.** In complex diseases different genotypic perturbations of the cellular system often lead to the same phenotype. While characteristic genomic alterations in many cancers exist, other combinations of genomic perturbations potentially lead to the same disease, dysregulating important pathways of the cellular system. In this study, we developed novel computational methods to identify dysregulated pathways and their direct causes in individual patients or patient groups. Specifically, we introduced efficient and powerful graph theoretic algorithms to identify such dysregulated pathways and their causal genes and applied our methods to a large set of glioma specific molecular data.

**Keywords:** Complex disease, genetic variations, copy number variation, biological pathway, graph theoretic algorithm, glioma.

## 1 Introduction

Complex diseases are typically caused by combinations of molecular perturbations that might vary strongly in different patients, dysregulating the same components (or pathways) of a cellular system (review [1]). For example, recent studies reported mutations, leading to dysregulated axon-guidance pathway genes in Parkinson Disease [2] and a set of genes, causing a possible disruption of neural activity-dependent regulation in autism [3] while diseases with similar phenotypes often are caused by mutations in functionally linked genes [4]. In recent years, whole-genome gene expression sets are increasingly used to search for markers, allowing the diagnosis of diseases or classifying their subtypes [5-11]. Several approaches combined expression measurements with various types of direct or indirect pathway information, obtaining improved disease classification [12-15], prioritization of disease associated genes [16-18], and identification of disease specific dys-regulated pathways [19]. Furthermore, considerable efforts towards integrated level approaches for uncovering disease causing genes ([20]; review [21]) and elucidation of relations between variability in gene expression and genotype (review [22]) have been recently made. In particular, Tu *et al.* implemented a random walk approach (used also in [16, 17]) to infer regulatory pathways [23] in yeast. Suthram *et al.* [24] further improved this approach by using the analogy between random walks and current flow in electric networks.

While previous methods allowed valuable insights into the modular nature of diseases by elucidating affected genes and pathways, they did not attempt to provide a

genome-wide view on possible causes of such dysregulation. Since biological pathways interact with each other in a variety of ways, one has to go beyond studying associations between individual disease genes and genotype variations in order to fully understand the complex disease network. In this paper, we address this challenge and present the first genome-wide approach to simultaneously determine dysregulated pathways and their putative direct causes/factors in individual patients and/or patient groups. As a model system we utilized gene expression and genomic profiles of human glioma patients, aggressive forms of human brain cancers that are characterized by genomic regions that are largely affected by copy number alterations such as amplifications, homozygous and heterozygous deletions as well as allelic imbalances such as loss of heterozygosity (LOH) and gene conversions [25]. Deletions of chromosomal areas, that contain tumor suppressor genes, can cause faulty regulation of important cell cycle processes, resulting in malignant cellular proliferation. Gene amplifications can promote over-expression of genes involved in cell proliferation and survival. Indeed, gene expression profiles allow a further classification of glioma subtypes [26]. While copy number variation and gene expression data in glioma provide opportunities to test our approach, our method also can be applied to other disease systems where genetic variations play a causal role.

Summarizing our contributions, (i) we developed a multi-step meta-analysis framework, integrating various types of data to facilitate the genome-wide discovery of disease causal genes and dys-regulated pathways. (ii) since our method models information flow in a biological network as a current flow in an electric circuit, and we needed to solve this problem for large number of randomized networks to provide a measure of statistical significance, we developed several optimization techniques leading to current flow algorithms fast enough to run efficiently on networks of the size of human interactome. (iii) We formulated the problem of finding a minimal set of putative causal genes for a given set of samples as a variant of a weighted multi-set cover problem and presented an efficient algorithm for this problem. (iv) We developed algorithms for two models of interaction networks. In our basic model, we used undirected edges allowing algorithmic efficiency and robustness with respect to noise in the network. In the refined model we utilized additional information such as the types and directions of interactions and developed a simple heuristic approach that allows us to solve the refined version of the problem on a large human interaction network. (v) Finally, we applied our method to genomic and gene expression data sets of glioma patients. Our approach returned causal genes, target genes, and pathway hubs that included a high proportion of known oncogenes, indicating the power of our approach.

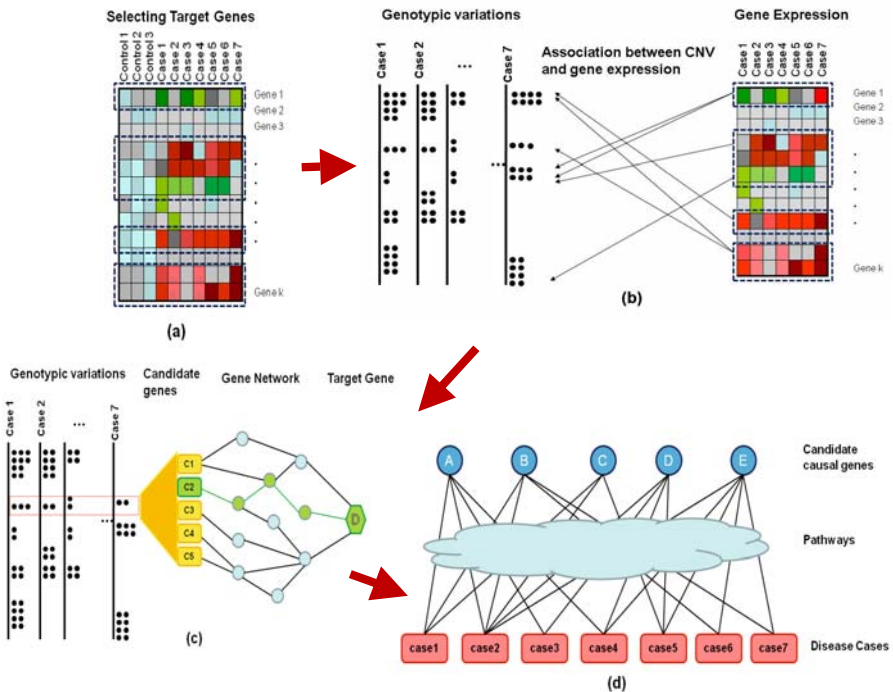
## 2 Methods

We developed a novel multistep algorithm to identify causal genes and associated dysregulated pathways by integrating several levels of analysis and data, including gene expression, genomic alterations and molecular interactions. Specifically, we aim to identify pathways, starting from genes that are located in areas of genomic alterations in human gliomas to potential target genes by following molecular interactions such as protein-protein interactions, phosphorylation events and protein-transcription factor interactions. We first describe the outline of the algorithm in Section 2.1 and

discuss the details of each step in the following subsections. Following the description of our basic approach, refinements of the method and corresponding results are presented as well.

### 2.1 Overview of Our Methods

Given a set of disease cases  $s_1, s_2, \dots, s_n$ , each disease case  $s_i$  is characterized by a genotype profile  $\{g_{i1}, g_{i2}, \dots, g_{ij} \dots\}$  where  $g_{ij}$  represents the genotype of case  $s_i$  in locus  $l_j$ . In our analysis, we utilized copy number alterations as genotypic information and gene expression profiles of each disease case,  $\{e_{i1}, e_{i2}, \dots, e_{ik} \dots\}$  where  $e_{i,k}$  represents the gene expression of case  $s_i$  for gene  $g_k$ . The outline of algorithm is illustrated in Fig. 1. In the first step, we identify a representative set of genes that are differentially expressed in disease cases (*target genes*), comparing gene expression profiles of disease samples and non-disease control cases (Fig. 1a) in the first step. The problem of identifying sets of representative target genes is related to finding disease markers and disease classification schemas. In our approach, we formulate the problem as a multi-set cover problem, determining a set of target genes with a greedy



**Fig. 1.** Schematic outline of the algorithm. (a) Selecting a representative set target genes are differentially expressed in disease cases compared to controls and are selected using a variant of multi-set cover approach. (b) Initial association of selected target genes with variations in genetic loci. (c) Identification of candidate casual genes using a network of molecular interactions. (d) Selection of a final set of causal genes and identification of pathway genes.

**Algorithm 1. Target gene selection**

1. for each pair of gene  $g_k$  and sample  $s_i$ , find  $p$ -value  $p(s_i, g_k)$  of gene expression data compared to those of non-disease samples.
2. create a multi-set cover instance  $SC = \{S, \Gamma, \alpha, \beta\}$
3.  $U =$  a set of cases covered less than  $\alpha$ .
4.  $TG =$  a set of selected genes
5.  $\Gamma$  is a collection of subsets  $C(g_k) \in S$
6. repeat the following until  $|U| \leq \beta$ 
  - 6.1. select a gene  $g_k \notin TG$  with maximum  $|U \cap C(g_k)|$
  - 6.2. include the selected gene in  $TG$
  - 6.3. update  $U$

approach. Informally, we ensure that each case is associated with (or *covered* by) some minimal number of selected differentially expressed genes and maximizing the overlap between these covers at the same time (see Section 2.2 for details). In the following,  $TG = \{tg_1, tg_2, \dots, tg_m\}$  denotes the set of target genes selected in this step.

In the next step (Fig. 1b and Section 2.3), we search for potential causes of the differential expression of target genes  $TG$ , utilizing an eQTL analysis that uncovers correlations between gene expression variation of target genes  $TG$  and copy number variation of loci. Since genomic data in neighboring regions tend to be highly correlated, we first choose a subset of representative loci (i.e. *tag loci*), significantly reducing computational costs and alleviating problems of multiple hypothesis testing. Note that the eQTL analysis in this step only provides putative eQTL regions that are associated with each target gene. Such regions are usually large and may include many false positives due to multiple testing issues and high correlations between neighboring markers.

We therefore further investigate the regions to identify causal genes and consider a gene to be more likely to be causal if there exists a path in the underlying interaction network that connects the causal gene with the corresponding target gene (Fig. 1c). In our analysis, we adopt a variant of a current flow algorithm [24], modeling the problem of finding a pathway through the network of molecular interactions as current flow in an electric circuit. In this way we find a set of candidate causal genes  $CG(i) = \{cg_{i,1}, cg_{i,2}, \dots, cg_{i,m}\}$  for each target gene  $tg_i$  and estimate the corresponding statistical significance with a permutation test (Section 2.4).

In the final step, we select a set of causal genes that explain the disease cases (Fig. 1d). We define a gene  $g_k$  as *causal* (i.e. explains a disease case  $s_i$ ) if the locus that includes the gene has a copy number alteration in case  $s_i$  while differentially expressed disease genes exist in the sample which are associated with the causal gene  $g_k$ . Given a set of candidate causal genes and disease cases, we formulate this problem as a variant of a weighted multi-set cover problem to find a minimum set of causal genes explaining (almost) all disease cases (Section 2.5). In addition, we find dysregulated pathways describing the flow of information between each pair of causal and target gene (Section 2.6).

Compared to molecular interactions in yeast [24], sets of interactions in human are several order of magnitudes larger, more noisy, and less complete. Furthermore, tissue specific interaction networks are not available that we expect to be rewired in the

disease stages. We consider all interactions as undirected in the basic version of our approach, increasing robustness with respect to missing information and mitigating algorithmic complexity issues. Since such tweaks are simplifications, we also developed a heuristic that accounts for the directions of phosphorylation events and protein-DNA interactions for biologically more accurate results (Section 2.7).

## 2.2 Selecting Target Genes

First, we identify genes that are differentially expressed in the disease cases compared to the non-disease controls in each case. Specifically, we normalized gene expression values as a Z-score, utilizing mean and standard deviation of gene expression values in the non-disease control cases. We consider a gene *differentially expressed* if the normalized gene expression value of the gene has a significant p-value in the given case using a Z-test.

Mapping all differentially expressed genes to all loci (see Section 2.3 for more detail) would not only require expensive computational cost but also suffer from multiple testing issues. Therefore, we choose a representative set of target genes for further analysis so that a sufficient number of differentially expressed genes are selected for each case. We formulate the problem of selecting target genes as a *minimum multi-set cover problem*: We construct a multi-set cover instance  $SC = \{S, \Gamma, \alpha, \beta\}$  where  $S$  is a set of cases.  $\Gamma$  is a collection of subsets  $C(g_k) \subseteq S$  for each gene  $g_k$  such that  $C(g_k)$  includes all cases for which gene  $g_k$  has a significant p-value.  $\alpha$  represents the number of times that a case needs to be covered, and  $\beta$  is the maximum number of outliers. In other words, all but  $\beta$  cases need to be covered at least  $\alpha$  times in the output cover.

### Algorithm 2. eQTL mapping

1. For each chromosome  $chr$ , let  $L_{chr}$  be the set of loci on the chromosome, sorted in increasing order of their genomic locations. Run the following for each chromosome.
  2.  $tl = L_{chr}[0] \setminus \setminus$  the first locus
  3. Add  $tl$  to  $TL \setminus \setminus$  TAG loci
  4. Consider loci in the sorted order
    - 4.1. if  $corr(tl, L_{chr}[i]) \leq \theta_{TL}$  ( $i$ : the current index,  $corr(x, y)$ : correlation coefficient)
      - $right(tl) = L_{chr}[i-1] \setminus \setminus$  set the right boundary of the old tag locus
      - $tl = L_{chr}[i]$  and include  $tl$  to  $TL \setminus \setminus$  select a new tag locus
      - Consider loci in reverse sorted order starting from  $j = i-1$
      - if  $corr(tl, L_{chr}[j]) \leq \theta_{TL}$
      - $left(tl) = L_{chr}[j+1] \setminus \setminus$  set the left boundary of the new tag
      - Go to 4.1
5. For each target gene  $dg_i$ 
  - 5.1.  $TL(i) = \emptyset$
  - 5.2. For each tag locus  $tl_j$ 
    - 5.2.1. Run linear regression between  $E(dg_i)$  and  $CN(tl_j)$  and compute  $p(dg_i, tl_j)$
    - 5.2.2. If  $p(dg_i, tl_j) < \theta_{eqtl}$   
Include  $tl_j$  to  $TL(i)$

The problem to choose a minimum number of genes, satisfying the constraints is NP-hard, prompting us to design a greedy algorithm. The pseudocode of the corresponding algorithm is shown in Algorithm 1.

### 2.3 eQTL Mapping

We utilize a set of loci  $L = \{l_1, l_2, \dots, l_m\}$  where each locus  $l_i$  is characterized by the corresponding copy number  $cn_{i,j}$  in each case  $j$ ,  $CN_i = \{cn_{i,1}, cn_{i,2}, \dots, cn_{i,n}\}$ . Since copy numbers of nearby loci tend to be highly correlated we can significantly reduce the number of loci by performing local clustering, allowing us to obtain a smaller set of tag loci. In Algorithm 2 we present the pseudocode of the tag loci selection algorithm where  $TL = \{tl_1, tl_2, \dots, tl_m\}$  is a set of tag loci and  $R(tl_k) = [left(tl_k), right(tl_k)]$  is the correlated genomic region of around each tag locus  $tl_k$ . Such regions include all consecutive loci including  $tl_k$ , ensuring that the Pearson’s correlation coefficient of  $CN_k$  and  $CN_i$  at any locus  $l_i$  in the region is  $> \theta_{TL}$ . Tag loci and associated regions can be computed in linear time. Highly correlated regions will be investigated in later steps to identify causal genes. Note that according to the algorithm, adjacent regions may overlap and a gene may belong to more than one region.

Given  $TL = \{tl_1, tl_2, \dots, tl_m\}$ , we identify candidate loci by associating copy number alteration with expression profiles of target genes. Given a set of target genes  $TG$  and tag loci  $TL$ , we calculate p-values  $p(tg_i, tl_j)$  by a linear regression between the expression values of gene  $tg_i$ ,  $E(tg_i)$ , and copy numbers of tag locus  $tl_j$ ,  $CN(tl_j)$  of all cases. For each target gene  $tg_i$ ,  $TL(i) \subseteq TL$  includes all tag loci with  $p(tg_i, tl_j) < \theta_{eqtl}$ . We consider a tag loci  $tl_j$  associated with  $tg_i$  if  $tl_j \in TL(i)$ .

### 2.4 Identifying Candidates Causal Genes

For each pair of a target gene  $tg_i$  and an associated tag locus  $tl_j \in TL(i)$ , we identify candidate causal genes in the region of the corresponding locus  $R(tl_j)$ . Inspired by the work of Suthram et al. [24], we adopt a variant of a current flow algorithm. For a given tag locus  $tl_j$  we first identify a set of genes  $C(tl_j)$  that are located in the corresponding locus region  $R(tl_j)$ . Using a network of molecular interactions, including protein-protein, protein-DNA interactions and phosphorylation events, we create an electric circuit, connecting potential causal genes in  $C(tl_j)$  and the target gene in question and compute the current flow from the target gene to its potential causal genes. The conductance of each interaction edge is given as a function of gene expression

**Algorithm 3. Selecting candidate causal genes**

1. For each disease gene  $dg_i$
2.  $CG(dg_i) = \emptyset$ 
  - 2.1 For each tag locus  $tl_j \in TL(i)$  and associated region  $R(tl_j)$ 
    - 2.1.1. Compute  $C(tl_j)$ , a set of genes located in  $R(tl_j)$
    - 2.1.2. Construct an electric circuit and compute current to each gene in  $C(tl_j)$
    - 2.1.3. Compute current in random networks and p-value for each gene in  $C(tl_j)$
    - 2.1.4.  $CG(dg_i) = CG(dg_i) \cup \{g \in C(tl_j) \mid p\text{-value}(g) \leq \theta_{current}\}$

correlation of the genes at the endpoints of edges and the target gene. The current flow is obtained by solving a system of linear equations as described in Section 2.4.1. Due to the large size of the human interaction network, standard software packages cannot solve the underlying system. In Section 2.4.2, we present optimization techniques to compute the solution and describe ways to estimate empirical p-values for each pair of target gene and tag locus given the solution of the linear system in Section 2.4.3. For each target gene  $tg$ , we find a set of candidate causal genes,  $CG(tg)$ , if  $p \leq \theta_{current}$  (Algorithm 3).

**2.4.1 Linear System**

The current flow algorithm is based on the well-known analogy between random walks and electronic networks where the amount of current entering a node or an edge in the network is proportional to the expected number of times a random walker will visit the node or edge. Let  $G = (N, E)$  represent a gene network where  $N$  is a set of genes and  $E$  is a set of molecular interactions. Let vector  $I = [I(e) \text{ for } e \in E]$  denote current passing through the edges and  $V = [V(n) \text{ for } n \in N]$  denote variables for voltage at the nodes. For a given tag locus, let  $C$  be the set of candidate genes located in its genomic region. Vector  $X = [X[c] \text{ for } c \in C]$  denotes the current leaving the candidate genes. For each edge  $e$ , we compute the weight  $w(e) = (corr(e[0], tg) + corr(e[1], tg))/2$ , which represents the conductance of the edge. Ohm’s law is defined as

$$Id * I + P * V = 0 \tag{1}$$

where  $Id$  is an  $|E| \times |E|$  identity matrix, and  $O$  is a zero matrix.  $P$  is an  $|E| \times |N|$  matrix and  $P(e, n) = w(e)$  if  $n = e[1]$ ,  $-w(e)$  if  $n = e[0]$ , and 0 otherwise. Kirchoff’s current law is

$$Q * I + R * X = T \tag{2}$$

where  $Q$  is an  $|N| \times |E|$  matrix, and  $Q(n, e) = 1$  if  $n = e[0]$ ,  $-1$  if  $n=e[1]$ , and 0 otherwise.  $R$  is an  $|N| \times |C|$  matrix where  $R(n, c) = 1$  if  $n = c$  and 0 otherwise.  $T$  is an  $|N| \times 1$  vector where  $T(n) = 1$  if  $n$  is the target gene  $tg$ , and 0 otherwise.

Finally, we set the voltage of all genes in  $C$  to be 0 so that all current flows into the candidate genes and there is no current flow between them, defined as

$$S * V = 0 \tag{3}$$

where  $S$  is a  $|C| \times |N|$  matrix and  $S(c, n) = 1$  if  $n=c$  and 0 otherwise. We are interested in the total current passing through each gene in  $C$  by solving the linear equations (1)-(3).

**2.4.2 Optimization**

The most computationally expensive part in our algorithm is the computation of the solution of the current flow. Since there exist dozens to hundreds of associated loci with  $\theta_{eqtl} = 0.01$  for each gene straightforward approaches take about several hours to days to compute the solution on the NCBI computing cluster. Furthermore, the size of linear system is  $O(|E|^2)$  due to the equation (1) and therefore, computations increasingly get intractable since the estimation of statistical significance needs additional

computation to solve the linear systems for a large number of randomized networks, prompting us to optimize the computation in several ways.

First, we combine eqs. (1) and (2) to reduce the size of matrix. By (1) we replace  $I$  in (2) with  $-P*V$ , and obtain

$$-Q*P*V + R*X = T \quad (4)$$

leaving us to solve the linear system (3) and (4), which reduces the size of linear system to  $O(|M|^2)$ .

The second optimization is to compute an inverse matrix using matrix decomposition [27] and utilize the fact that all tag loci associated with the same target gene have a common matrix  $Q*P$ . Considering the matrix representation of our linear system

$$\begin{bmatrix} -Q*P & R \\ S & O \end{bmatrix} \begin{bmatrix} V \\ X \end{bmatrix} = \begin{bmatrix} T \\ O \end{bmatrix} \quad (5)$$

the solution can be obtained by computing a matrix inversion as

$$\begin{aligned} \begin{bmatrix} V \\ X \end{bmatrix} &= \begin{bmatrix} -Q*P & R \\ S & O \end{bmatrix}^{-1} \begin{bmatrix} T \\ O \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} T \\ O \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} [T] \\ &= \begin{bmatrix} -\overline{QP}(Id - R*R*(S*\overline{QP}*R)^{-1}*S*\overline{QP}) \\ (S*\overline{QP}*R)^{-1}*S*\overline{QP} \end{bmatrix} [T] \end{aligned} \quad (6)$$

where  $\overline{QP} = (Q*P)^{-1}$ . Note that  $\overline{QP}$  can be pre-computed and reused to compute  $[V^T X^T]$  for each associated tag locus. Even though solving linear system by computing inverse matrices typically takes more time than other methods, our algorithm requires the  $|M| \times |M|$  matrix inversion operation only once for each target gene, allowing us to fast compute solutions for all loci associated to a target gene.

### 2.4.3 Computing Empirical p-Values

Given the solution of the linear system, an empirical p-value for each pair of target gene and tag locus is estimated by generating 50 random networks, swapping edges while preserving node degrees. Assuming that each edge has a unit conductance, we run the current flow algorithm in each random network for the same set of genes and compute the amount of current flowing into each gene located in the tag locus. A normal distribution was fitted to the current values in the random network and empirical p-values are computed with a Z-test.

## 2.5 Selecting Final Set of Causal Genes

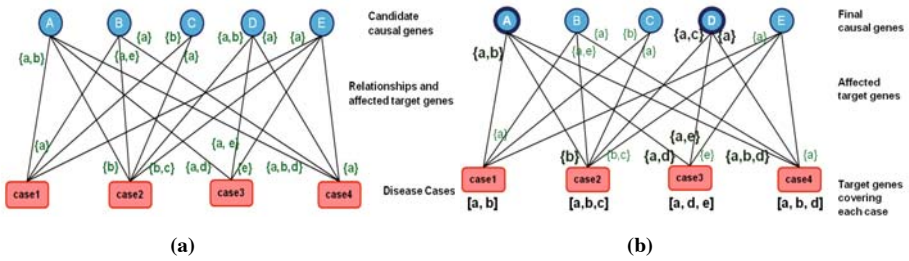
One of our primary goals is to identify a set of causal genes that explain (almost) all disease cases. Given the candidate causal genes affecting target genes (computed by Algorithm 3) and the copy number variation of genotypes we identify a handful of common causal genes that explain the disease cases. A target gene  $tg_j$  is affected by a causal gene  $cg_k$  if  $cg_k \in CG(tg_j)$ , i.e., the current flow from  $tg_j$  to  $cg_k$  has a significant p-value. We formally define a causal gene explaining a case as follows:

**Definition 1.** A causal gene  $cg_k$  explains a case  $s_i$  iff (i) the tag locus including the gene has copy number alterations in case  $s_i$  and (ii) there exists a set of target



$gene(s)$ ,  $D(cg_k, s_i)$ , which are affected by  $cg_k$  and differentially expressed in case  $s_i$ . The weight between a causal gene and a case,  $w(k, i)$  is defined as the size of  $D(cg_k, s_i)$ .

A bipartite graph  $B(C, S)$  between a set of candidate causal genes  $C$  and disease cases  $S$  can be constructed by adding edges between gene  $cg_k$  and  $s_i$  if and only if gene  $cg_k$  explains a case  $s_i$  (Fig. 2a). For a subset of candidate causal genes  $C_0$  and a case  $s$ , let  $W(C_0, s)$  be the total number of target genes covering  $s$  by the genes in  $C_0$ ,  $W(C_0, s) = |\bigcup_{c \in C_0} D(c, s)|$ . A case is explained if the total weight covering the sample exceeds a certain threshold. As in the preprocessing in the first step, we want to explain all cases (allowing a few outliers) with minimum number of causal genes. In Fig. 2b, two causal genes  $\{A, D\}$  are selected covering each case (except case 1) at least 3 times.



**Fig. 2.** Selection of final causal genes. (a) In an example of a bipartite graph between candidate causal genes and disease cases each edge is labeled with the associated set of target genes that are affected by the causal gene and differentially expressed in the corresponding disease case. (b) The set of selected final causal genes  $\{A, D\}$  covers each disease case at least three times (except case 1).

The problem can be formulated as a variant of *minimum weighted multi-set cover problem*. Consider an instance  $WSC = \{B, \gamma, \delta\}$  where  $B$  is a weighted bipartite graph between  $C$  and  $S$ . We want to choose a subset of genes  $C'$  from  $C$  such that for each case except  $\delta$  cases,  $W(C', s) \geq \gamma$ . Since a very simple version of the multi-set cover problem (unweighted without outliers) is NP-hard, we designed an algorithm, using a greedy approach to choose a subset of genes. Repeatedly, we compute the total weight that can be covered by choosing a gene and select a gene with maximum total weight until we meet the stop criterion (See Algorithm 4 for the details).

### 2.6 Identifying Dysregulated Pathways

Finally, we determine dysregulated pathways. Let  $r(c)$  be the regions that contain a causal gene  $c$ . Since regions may overlap, a gene can be part of more than one region. Let  $r_{max}(c, d)$  and  $tl_{max}(c, d)$  be the region and tag locus that harbor gene  $c$  and have the most significant p-value among all the current flow solutions from  $d$  to regions in  $r(c)$ .

**Algorithm 4. Selection of final causal genes**

1. Create a weighted multi-set cover instance  $WSC = \{B, \gamma, \delta\}$
2.  $U =$  a set of cases covered less than  $\gamma$ .
3.  $MCG =$  a set of selected causal genes
4. Repeat the following until  $|U| \leq \delta$ 
  - 4.1. Select a gene  $g_k \notin MCG$  with maximum

$$W(g_k, U) = \sum_{s \in U} (W(MCG \cup \{g_k\}, s) - W(MCG, s))$$

- 4.2. Include the selected gene in  $MCG$
- 4.3. Update  $U$

Utilizing a current flow solution  $Sol(d, tl_{max}(c, d))$  from  $d$  to  $tl_{max}(c, d)$  we determined a path from  $c$  to  $d$  by defining a maximum current path from  $d$  to  $c$  as a simple path  $P(d, c) = (d, g_1, g_2, \dots, c)$  such that  $\min_{g_i \text{ in } P(d, c)} I(g_i)$  is maximized where  $I(g_i)$  is the total current passing through the gene  $g_i$ . We compute a path for each pair of a final causal gene and a target gene affected by the underlying causal gene.

## 2.7 Algorithm Refinements

In our basic algorithm, we considered all interactions as undirected. Furthermore, we assumed that each interaction has a regulatory effect on a target gene. Aiming to obtain more biologically meaningful results, we modify the basic version of our algorithms on two accounts. First, we assume that direct regulation activity on the expression of disease is mediated by transcription factors only. Therefore, we implemented a version of the algorithm, determining paths where target genes interact with transcription factors only. Second, we account for directions of protein-DNA interactions and phosphorylation events, allowing us to interpret the results in the context of information flow in the cell. One way to obtain current flow in directed networks is to solve a linear programming [24]. However, such an approach is computationally extremely costly given the large size of our human molecular interaction network. We address this problem by a simple heuristic approach by searching for edges that are used in opposite direction after solving the linear system. Removing these edges we solve the linear system for the remaining edges again and repeat this process until only a small number of directed edges are used incorrectly.

## 3 Data

### 3.1 mRNA Data Treatment

We utilized 321 patient and 32 non-tumor control samples collected from the NCI-sponsored Glioma Molecular Diagnostic Initiative (GMDI) which were profiled using HG-U133 Plus 2.0 arrays. Arrays were normalized at the PM and MM probe level with dChip [26, 28]. Using the average difference model to compute expression values, model-based expression levels were calculated with normalized probe level data, and negative average differences (MM > PM) were set to 0 after log-transforming expression values [26]. Accounting for weak signal intensities, all probesets with

more than 10% of zero log-transformed expression values were removed. Representing each gene, we chose the corresponding probeset with the highest mean intensity in the tumor and control samples.

### 3.2 Determination of Copy Number Alterations

The same 321 patient and 32 non-tumor control samples were hybridized on the Genechip Human Mapping 100K arrays, and copy numbers were calculated using Affymetrix Copy Number Analysis Tool (CNAT 4). After probe-level normalization and summarization, calculated log<sub>2</sub>-transformed ratios were used to estimate raw copy numbers. Using a Gaussian approach, raw SNP profiles were smoothed (> 500 kb window by default) and segmented using a Hidden Markov Model approach [25, 29, 30]. Considering alterations of copy numbers (CN), we define an amplification if log<sub>2</sub> CN -1 > 0.1 and a deletion if log<sub>2</sub> CN -1 < -0.1.

### 3.3 Interaction Network

We utilized human protein-protein interaction data from large-scale high-throughput screens [31-33] and several curated interaction databases [34-37] totaling 93,178 interactions among 11,691 genes. As a reliable source of experimentally confirmed protein-DNA interactions, we used 6,669 interactions between 2,822 transcription factors and structural genes from the TRED database [38]. As for phosphorylation events between kinases and other proteins we found 5,462 interactions between 1,707 human proteins utilizing networKIN [39, 40] and phosphoELM database [41]. Pooling all interactions we obtained a network of 11,969 human proteins that are connected by 103,966 links.

## 4 Results

### 4.1 Target Genes

A gene is defined to be differentially expressed for a case if p-value is less than 0.01 when a standard normal distribution is fitted to the control. In the basic model, we solved a multi-set cover instance  $SC = \{S, T, 30, 15\}$  using Algorithm 1, and obtained 73 target genes. The selected target genes and their expression patterns are shown in Table 1. Four genes, CD200R1, CH25H, GPR27 and WNT6 are not considered in the later analysis due to the weak signal intensities of gene expression.

Only 25 target genes that were selected in the basic model have transcription factors. For the refined model, we only consider genes with transcription factors and selected 77 target genes using a multi-set cover ( $\alpha = 25, \beta = 15$ ). All 25 genes selected in the basic model are also selected in the refined model (Table 1). Since only 25-30% of genes in the network have known transcription factor, we have less coverage (25 times each case) even though more target genes are selected in the refined model. Searching literature and manually examining our 25 genes in the common set using AceView [42] we found all but 6 genes, ATN1, PIGQ, C20orf108, RNASE2, TBL2 and UBN1 to be cancer-related, while one of the remaining four, ATN1, is related to brain diseases.

**Table 1.** Target, hub and causal genes found using the basic and refined model. The common set contains genes found in both models. As for the color-code of target and hub genes, red genes are up-regulated and green genes are down-regulated. Causal genes are marked red if they were found in amplified region and green if they were found in deleted genomic regions.

	Basic model only	Common set	Refined model only
Target Genes	BAT2, BOC, CAMK2N1, CD200R1, CH25H, CUL7, DTNA, ELAVL1, EPB41L1, FAM60A, FKBP5, FLJ21865, GPR27, KCNAB1, KIAA1107, KIF5C, LIN7B, LRP1, LSM8, MAML2, MAPK8IP3, MAPKAPK3, MAZ, MTHFD2, NES, NME4, NRP2, NTSDC2, NUAK1, OBSL1, PDE1A, RAMP1, RCN1, REPS2, RND2, RPGR, SIGLEC1, SOX7, STXPB6, TAGAP, TGFB11, TNK2, UBE2D4, WNT6, ZDHHC21, ZFYVE27, ZNF212, ZNF365	ATN1, BLM, BTG1, C20orf108, CD80, CDK4, EGFR, ESRRA, F2R, FAS, LAMC1, LPL, MAD2L1, MADD, MFNG, PDGFRA, PIGQ, PPP2R2C, PRRX1, RNASE2, SOX11, TBL2, TGFB2, TSPAN6, UBN1	ABCA1, ANTXR1, BCL6, CCHCR1, CDK2, CEBPB, DDIT4, DECR1, DGKA, DUSP1, EPHX1, FASN, FLNA, FOS, GNAI2, GOT1, GPX3, HEY1, HHEX, ID4, IL18BP, JUNB, KIAA0020, LDLR, MAP1A, MAP2K6, METTL1, NEDD9, NR4A2, OXTR, PECAM1, PKD1, PNRC1, RBBP8, RIPK2, SESN1, SHMT2, SLC2A1, SNRPG, SOX2, STMN1, SULF2, SYK, TIMP3, TP53, TPRKB, UHRF1, USF2, VAMP1, VAMP2, VEGFB, ZEB1
Final Causal Genes	CDC2, EGFR, FANCL, GBAS, ITGB1, MTAP, NDRG1, PRKD1, SFRS7, SNRPD2	AKAP6, DDB1, DPP10, GLO1, POU2F1, PTEN, RPL13A, SFRS11, SLIT1, SNRPA, SNRPB, TOP1, TRRAP, WDR8, YBX1, ZNF107	CDC42, DDX54, EIF4H, FBXO25, GRB10, NEDD8, NP, PCBP1, POLR2F, POLR2J, PPP3R1, PTK2, RPS24, TNFRSF1B, TRPS1, UBA52, USP7
Hub genes	CDK2, GABARAPL2, GRB2, HIPK3, NP, PRKCB1, RPS6KA3, SNUPN, TP53, UBA52, YWHAG	E2F4, E2F1, PRKCA, SP1, RPS27A, GSK3B, ARR2, MYC, RELA	AR, CDC2, CDK7, CREBBP, DKFZp586M0622, EP300, HIF1A, MYB, RB1, SMAD3, TFAP2A

## 4.2 eQTL Mapping

Among 50K SNPs, 905 tag loci have been selected with a correlation threshold = 0.9 in Algorithm 2. We performed linear regression between copy number alterations of the tag loci and target genes and chose tag loci with  $p < 0.01$ . On average, we found 104 tag loci per target gene, while transforming growth factor TGFB2 led with 233 associated loci.

## 4.3 Causal Genes

We applied Algorithm 3 for all pairs of target genes and associated loci. The number of genes located in each region varied from 0 to several dozens, and therefore the amount of current that flows to genes cannot be compared directly among different loci to prioritize the genes. For each locus and a set of genes in the associated region, we only consider genes receiving current of at least 70% of the maximum current among all genes in the region. We then use a permutation method to obtain empirical p-values and select candidate causal genes for each target gene if the empirical, gene

specific  $p \leq 0.05$ . On average, each target gene has 56 candidate causal genes, and the final causal genes are selected using Algorithm 4.

For the basic model, we created a weighted multi-set cover instance  $\{B, 25, 20\}$  and selected 26 *final causal genes*. To evaluate the performance of our method, we compiled 166 genes related to Glioma (which we call GLIOMA\_GENES) using AceView<sup>1</sup>. Three genes (PTEN, EGFR, MTAP) in the final causal genes are also included in GLIOMA\_GENES (p-value = 0.004). We note that disease associated genes in AceView are automatically collected, using a literature-mining algorithm, and may not be comprehensive. For example, our causal gene set includes GBAS (glioblastoma amplified sequence, a gene that is reported amplified in approximately 40% of glioblastomas [43]). By manually searching literature, we also found that more than half of causal genes in the basic model (14 genes) are associated with cancer. In the refined model, 33 final causal genes are selected using the weighted multi-set cover instance  $\{B, 25, 30\}$ . Among 17 genes that were not included in the basic model, 10 genes were found to be cancer related. In particular, PTK2 (also called FAK) is known to be upregulated in anaplastic astrocytoma and glioblastoma and may play a role in the promotion of glioblastoma cell proliferation, survival and migration [44]. The selected genes are listed in the second row of Table 1. Genes are marked as red for copy number amplification and green for deletion<sup>2</sup>.

Interestingly, among the predicated causal genes are two classical “antagonist” genes: PTEN, a tumor suppressor gene and EGFR, a tumor activator. Both genes, located at chromosome 10 and 7 respectively, are frequently affected by copy number alterations. Indeed, we found that the genomic profiles of some patients only show an increase of gene copy number in EGFR while some only have deletions in the PTEN region. In turn, some patients show both or even none of these two variations. However, there exists a large number of other cancer related genes among the identified causal genes that can serve as conduits to the cancer causing perturbations.

#### 4.4 Dysregulated Pathways

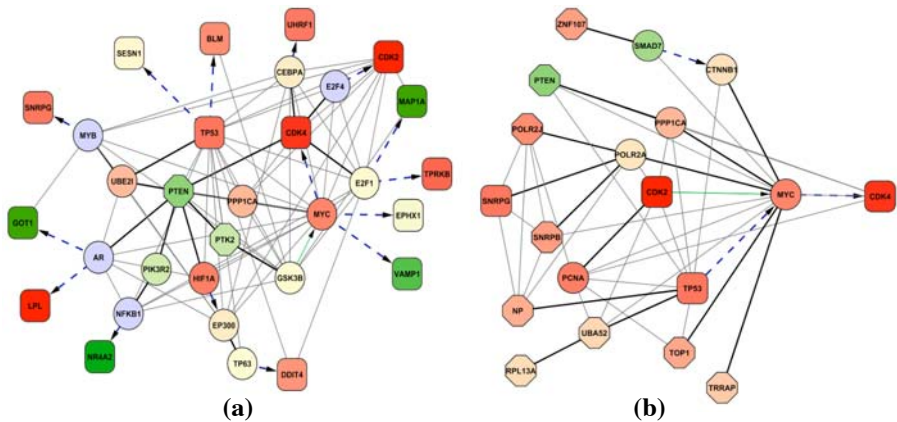
Maximum current paths have been computed between every pair of target and its causal gene as described in Section 2.6. The network combining all those paths includes 348 (311) nodes and 644 (691) interactions in the basic (refined) model. Several important cancer and brain disease related genes appear as intermediate hub nodes in the network. The top 20 genes with highest degree (“*hub genes*”) for each model are listed in the bottom row of Table 1. The list includes many cancer related genes MYC(78, 53), TP53(16, 57), CDK2(21, 22), E2F1(15, 18), PRKCA (18, 14), E2F4(9, 22), GSK3B(18, 12), AR(5, 22), RELA(7, 20), SP1(11, 17), and GRB2(13, 6) where the numbers inside the parenthesis indicate the degrees of the genes in the basic and refined model.

---

<sup>1</sup> 174 genes are listed as Glioma associated genes in Aceview. After removing genes with no valid expression data or not appearing in our gene network, we obtained 166 genes in GLIOMA\_GENES.

<sup>2</sup> In case that there are mixed samples with amplification or deletion in the genomic region, we performed binomial testing with p-value threshold 0.05.

Fig. 3a shows the subnet in the refined model where PTEN, which is a tumor suppressor that has mutations in a large number of cancer types, is considered as a causal gene and includes all paths to target genes that are affected by PTEN. Specifically, the PTEN network is significantly enriched for genes that are involved in several cancer related cellular processes such as cell cycle and ER overload response. In Fig. 3b, we show a subnetwork that revolves around CDK4 as a target gene that is connected to its causal genes. We find that the network is significantly enriched with genes that are involved in cell cycle processes and positive regulation of cell proliferation. Furthermore, an overwhelming majority of networks hubs is known to be associated with cancer.



**Fig. 3.** Subnetworks exemplifying dys-regulated pathways obtained with the refined algorithm. The color of each gene shows its expression level: red (up-regulated), green (down-regulated), yellow (neutral), and gray (no expression data). Rectangular and octagon nodes are the genes identified as target and final causal genes, respectively. Thick edges appear in the maximum current paths, and background edges are obtained by computing induced sub-graphs. The dotted edges are protein-DNA interactions and green edges represent phosphorylation. In (a), we show the sub-graph between PTEN and its affected target genes while in (b) we present the subnet between CDK4 and affecting causal genes.

## 5 Discussion

In this work we proposed the first approach for simultaneous identification of causal genes of diseases and dys-regulated pathways. In our approach, we started by identification of “seed” target genes by a vertex multicover approach. Followed by the identification of putative causal genes by a simple eQTL analysis, we refined the initial set by modeling and solving a current flow problem. The latter step additionally uncovered other prominent nodes in the network connecting causal genes to the representing target genes.

As such, pathways uncovered by the current flow algorithm can be considered as possible explanations for causes that lead to the phenotype we observed. Even though

our integrative approach allows us to uncover interesting regulatory subnetworks, many other sources of auxiliary information remain to be integrated. For example, our current approach might be extended with regulatory interactions provided by miRNAs as well as epigenetic interactions. As for the current status of molecular interactions, we acknowledge that the current network of protein interactions, protein-DNA interactions and phosphorylation events is incomplete and noisy. Despite these data specific problems augmenting eQTL evidence with pathway information resulted in a very powerful approach, allowing us to not only uncover potential disease genes, but also find intermediate nodes on molecular pathways that mediate information between causal genes and disease marker genes.

## References

1. Schadt, E.E.: Molecular networks as sensors and drivers of common human diseases. *Nature* 461, 218–223 (2009)
2. Lesnick, T.G., Papapetropoulos, S., Mash, D.C., Ffrench-Mullen, J., Shehadeh, L., de Andrade, M., Henley, J.R., Rocca, W.A., Ahlskog, J.E., Maraganore, D.M.: A genomic pathway approach to a complex disease: axon guidance and Parkinson disease. *PLoS Genet.* 3 3, e98 (2007)
3. Morrow, E.M., Yoo, S.Y., Flavell, S.W., Kim, T.K., Lin, Y., Hill, R.S., Mukaddes, N.M., Balkhy, S., Gascon, G., Hashmi, A., Al-Saad, S., Ware, J., Joseph, R.M., Greenblatt, R., Gleason, D., Ertelt, J.A., Apse, K.A., Bodell, A., Partlow, J.N., Barry, B., Yao, H., Markianos, K., Ferland, R.J., Greenberg, M.E., Walsh, C.A.: Identifying autism loci and genes by tracing recent shared ancestry. *Science* 321, 218–223 (2008)
4. Oti, M., Brunner, H.G.: The modular nature of genetic diseases. *Clin. Genet.* 71, 1–11 (2007)
5. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537 (1999)
6. Perou, C.M., Sorlie, T., Eisen, M.B., van de Rijn, M., Jeffrey, S.S., Rees, C.A., Pollack, J.R., Ross, D.T., Johnsen, H., Akslen, L.A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S.X., Lonning, P.E., Borresen-Dale, A.L., Brown, P.O., Botstein, D.: Molecular portraits of human breast tumours. *Nature* 406, 747–752 (2000)
7. Ramaswamy, S., Ross, K.N., Lander, E.S., Golub, T.R.: A molecular signature of metastasis in primary solid tumors. *Nat. Genet.* 33, 49–54 (2003)
8. Nagasaki, K., Miki, Y.: Gene expression profiling of breast cancer. *Breast Cancer* 13, 2–7 (2006)
9. Thompson, M., Lapointe, J., Choi, Y.L., Ong, D.E., Higgins, J.P., Brooks, J.D., Pollack, J.R.: Identification of candidate prostate cancer genes through comparative expression-profiling of seminal vesicle. *Prostate* 68, 1248–1256 (2008)
10. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson Jr., J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403, 503–511 (2000)

11. van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H.: Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536 (2002)
12. Doniger, S.W., Salomonis, N., Dahlquist, K.D., Vranizan, K., Lawlor, S.C., Conklin, B.R.: MAPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biology* 4, R7 (2003)
13. Lee, E., Chuang, H.Y., Kim, J.W., Ideker, T., Lee, D.: Inferring pathway activity toward precise disease classification. *PLoS Comput. Biol.* 4, e1000217 (2008)
14. Keller, A., Backes, C., Gerasch, A., Kaufmann, M., Kohlbacher, O., Meese, E., Lenhof, H.P.: A novel algorithm for detecting differentially regulated paths based on Gene Set Enrichment Analysis. *Bioinformatics* (2009)
15. Chuang, H.Y., Lee, E., Liu, Y.T., Lee, D., Ideker, T.: Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* 3, 140 (2007)
16. Kohler, S., Bauer, S., Horn, D., Robinson, P.N.: Walking the interactome for prioritization of candidate genes. *Am. J. Human Genet.* 82, 949–958 (2008)
17. Vanunu, O., Sharan, R.: A propagation-based algorithm for inferring gene-disease associations. In: German Conference on Bioinformatics. LNI, vol. 136 (2008)
18. Wu, X., Jiang, R., Zhang, M.Q., Li, S.: Network-based global inference of human disease. *Mol. Sys. Biol.* 4, 189 (2008)
19. Ulitsky, I., Karp, R., Shamir, R.: Detecting Disease-Specific Dysregulated Pathways Via Analysis of Clinical Expression Profiles Research in Computational Molecular Biology, 347–359 (2008)
20. Schadt, E.E., Lamb, J., Yang, X., Zhu, J., Edwards, S., Guhathakurta, D., Sieberts, S.K., Monks, S., Reitman, M., Zhang, C., Lum, P.Y., Leonardson, A., Thieringer, R., Metzger, J.M., Yang, L., Castle, J., Zhu, H., Kash, S.F., Drake, T.A., Sachs, A., Lusk, A.J.: An integrative genomics approach to infer causal associations between gene expression and disease. *Nat. Genet.* 37, 710–717 (2005)
21. Sieberts, S.K., Schadt, E.E.: Moving toward a system genetics view of disease. *Mamm Genome* 18, 389–401 (2007)
22. Huang, Y., Zheng, J., Przytycka, T.: Discovery of regulatory mechanisms by genome-wide from gene expression variation by eQTL analysis. In: Lonardi, J.Y.C.a.S. (ed.) *Biological Data Mining*, pp. 205–228. CRC Press, Boca Raton (2009)
23. Tu, Z., Wang, L., Arbeitman, M.N., Chen, T., Sun, F.: An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics* 22, e489–e496 (2006)
24. Suthram, S., Beyer, A., Karp, R.M., Eldar, Y., Ideker, T.: eQED: an efficient method for interpreting eQTL associations using protein networks. *Mol. Syst. Biol.* 4, 162 (2008)
25. Kotliarov, Y., Steed, M.E., Christopher, N., Walling, J., Su, Q., Center, A., Heiss, J., Rosenblum, M., Mikkelsen, T., Zenklusen, J.C., Fine, H.A.: High-resolution global genomic survey of 178 gliomas reveals novel regions of copy number alteration and allelic imbalances. *Cancer Res.* 66, 9428–9436 (2006)
26. Li, A., Walling, J., Ahn, S., Kotliarov, Y., Su, Q., Quezado, M., Oberholtzer, J.C., Park, J., Zenklusen, J.C., Fine, H.A.: Unsupervised analysis of transcriptomic profiles reveals six glioma subtypes. *Cancer Res.* 69, 2091–2099 (2009)
27. Press, W., Teukowsky, S.A., Vetterling, W.T., Fannery, B.P.: *Numerical Recipes - The Art of Scientific Computing*. Cambridge University Press, Cambridge (2007)
28. Li, C., Wong, W.H.: Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc. Natl. Acad. Sci. U.S.A.* 98, 31–36 (2001)



29. Fridlyand, J., Snijders, A.M., Pinkel, D., Albertson, D.G., Jain, A.N.: Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis* 90, 132–153 (2004)
30. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A.J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J.Y., Zhang, J.: Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.* 5, R80 (2004)
31. Ewing, R.M., Chu, P., Elisma, F., Li, H., Taylor, P., Climie, S., McBroom-Cerajewski, L., Robinson, M.D., O'Connor, L., Li, M., Taylor, R., Dharsee, M., Ho, Y., Heilbut, A., Moore, L., Zhang, S., Ornatsky, O., Bukhman, Y.V., Ethier, M., Sheng, Y., Vasilescu, J., Abu-Farha, M., Lambert, J.P., Duewel, H.S., Stewart II, Kuehl, B., Hogue, K., Colwill, K., Gladwish, K., Muskat, B., Kinach, R., Adams, S.L., Moran, M.F., Morin, G.B., Topaloglou, T., Figeys, D.: Large-scale mapping of human protein-protein interactions by mass spectrometry. *Mol. Syst. Biol.* 3, 89 (2007)
32. Rual, J.F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G.F., Gibbons, F.D., Dreze, M., Ayivi-Guedehoussou, N., Klitgord, N., Simon, C., Boxem, M., Milstein, S., Rosenberg, J., Goldberg, D.S., Zhang, L.V., Wong, S.L., Franklin, G., Li, S., Albala, J.S., Lim, J., Fraughton, C., Llamasas, E., Cevik, S., Bex, C., Lamesch, P., Sikorski, R.S., Vandenhaute, J., Zoghbi, H.Y., Smolyar, A., Bosak, S., Sequerra, R., Doucette-Stamm, L., Cusick, M.E., Hill, D.E., Roth, F.P., Vidal, M.: Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 437, 1173–1178 (2005)
33. Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F.H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., Timm, J., Mintzlauff, S., Abraham, C., Bock, N., Kietzmann, S., Goedde, A., Toksoz, E., Droege, A., Krobitsch, S., Korn, B., Birchmeier, W., Lehrach, H., Wanker, E.E.: A human protein-protein interaction network: a resource for annotating the proteome. *Cell* 122, 957–968 (2005)
34. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INteraction database. *Nucleic Acids Res.* 35, D572–D574 (2007)
35. Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Liefink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorncroft, D., Zhang, Y., Apweiler, R., Hermjakob, H.: IntAct—open source resource for molecular interaction data. *Nucleic Acids Res.* 35, D561–D565 (2007)
36. Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S., Mahajan, S., May, B., Schmidt, E., Vastrik, I., Wu, G., Birney, E., Stein, L., D'Eustachio, P.: Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.* (2008)
37. Peri, S., Navarro, J.D., Kristiansen, T.Z., Amanchy, R., Surendranath, V., Muthusamy, B., Gandhi, T.K., Chandrika, K.N., Deshpande, N., Suresh, S., Rashmi, B.P., Shanker, K., Padma, N., Niranjana, V., Harsha, H.C., Talreja, N., Vrushabendra, B.M., Ramya, M.A., Yatish, A.J., Joy, M., Shivashankar, H.N., Kavitha, M.P., Menezes, M., Choudhury, D.R., Ghosh, N., Saravana, R., Chandran, S., Mohan, S., Jonnalagadda, C.K., Prasad, C.K., Kumar-Sinha, C., Deshpande, K.S., Pandey, A.: Human protein reference database as a discovery resource for proteomics. *Nucleic Acids Res.* 32, D497–D501 (2004)
38. Jiang, C., Xuan, Z., Zhao, F., Zhang, M.Q.: TRED: a transcriptional regulatory element database, new entries and other development. *Nucleic Acids Res.* 35, D137–D140 (2007)

39. Linding, R., Jensen, L.J., Ostheimer, G.J., van Vugt, M.A., Jorgensen, C., Miron, I.M., Diella, F., Colwill, K., Taylor, L., Elder, K., Metalnikov, P., Nguyen, V., Pasculescu, A., Jin, J., Park, J.G., Samson, L.D., Woodgett, J.R., Russell, R.B., Bork, P., Yaffe, M.B., Pawson, T.: Systematic discovery of in vivo phosphorylation networks. *Cell* 129, 1415–1426 (2007)
40. Linding, R., Jensen, L.J., Pasculescu, A., Olhovsky, M., Colwill, K., Bork, P., Yaffe, M.B., Pawson, T.: NetworKIN: a resource for exploring cellular phosphorylation networks. *Nucleic Acids Res.* 36, D695–D699 (2008)
41. Diella, F., Gould, C.M., Chica, C., Via, A., Gibson, T.J.: Phospho.ELM: a database of phosphorylation sites—update 2008. *Nucleic Acids Res.* 36, D240–D244 (2008)
42. Thierry-Mieg, D., Thierry-Mieg, J.: AceView: a comprehensive cDNA-supported gene and transcripts annotation. *Genome Biol.* 7(suppl.1), S12 11–14 (2006)
43. Wang, X.Y., Smith, D.I., Liu, W., James, C.D.: GBAS, a novel gene encoding a protein with tyrosine phosphorylation sites and a transmembrane domain, is co-amplified with EGFR. *Genomics* 49, 448–451 (1998)
44. Natarajan, M., Hecker, T.P., Gladson, C.L.: FAK signaling in anaplastic astrocytoma and glioblastoma tumors. *Cancer journal (Sudbury, Mass)* 9, 126–130 (2003)

# Incremental Signaling Pathway Modeling by Data Integration

Geoffrey Koh<sup>1,\*</sup>, David Hsu<sup>2</sup>, and P.S. Thiagarajan<sup>2</sup>

<sup>1</sup> Bioprocessing Technology Institute, Singapore, 117456, Singapore

<sup>2</sup> National University of Singapore, Singapore, 117417, Singapore

**Abstract.** Constructing quantitative dynamic models of signaling pathways is an important task for computational systems biology. Pathway model construction is often an inherently *incremental* process, with new pathway players and interactions continuously being discovered and additional experimental data being generated. Here we focus on the problem of performing model parameter estimation incrementally by integrating new experimental data into an existing model. A probabilistic graphical model known as the *factor graph* is used to represent pathway parameter estimates. By exploiting the network structure of a pathway, a factor graph compactly encodes many parameter estimates of varying quality as a probability distribution. When new data arrives, the parameter estimates are refined efficiently by applying a probabilistic inference algorithm known as *belief propagation* to the factor graph. A key advantage of our approach is that the factor graph model contains enough information about the old data, and uses only new data to refine the parameter estimates without requiring explicit access to the old data. To test this approach, we applied it to the Akt-MAPK pathways, which regulate the apoptotic process and are among the most actively studied signaling pathways. The results show that our new approach can obtain parameter estimates that fit the data well and refine them incrementally when new data arrives.

## 1 Introduction

To fully understand complex biological pathways, we must uncover not only the constituent elements—genes, proteins, and other molecular species—and their interactions, but also the *dynamics*, *i.e.*, the evolution of these interactions over time. One important goal of computational systems biology is to build quantitative models of pathway dynamics [12]. These models should not only capture our understanding of the underlying mechanisms, but also predict behaviors yet to be observed experimentally. A key challenge is to address the inherently incremental nature of the model construction process, as new pathway players and interactions are discovered and additional experimental data are generated. In this work, we address the problem of incrementally constructing pathway models as new data becomes available.

A signaling pathway is a network of biochemical reactions. To build a model, we need both the network structure and the parameters. Structure modeling captures the interdependencies among the molecular species, based on the reactions producing and

---

\* The work was done when G. Koh was a PhD student at the National University of Singapore.

consuming them. Parameter modeling determines the kinetic rate constants, initial conditions, *etc.* that govern the biochemical reactions. Here, we focus on parameter modeling, also called parameter estimation.

Parameter estimation for large signaling pathways is a well-known difficult problem, due to the need to search a high-dimensional parameter space and the lack of accurate data. Conventional parameter estimation algorithms fit an estimate of the parameters with all available experimental data and produce a single best estimate of the parameters (see [3] for a survey). When new data arrives, the entire procedure must be repeated afresh, in order to fit both the new and the old data well. This simplistic approach of recomputing the parameter estimate is undesirable. It does not take advantage of the earlier estimates. Furthermore, it may be not even be feasible, if the old data is not easily accessible. Often, many parts of the current model are obtained from external sources. For these “imported” parts, we have the estimated parameter values, but are unlikely to have access to the data used to produce these estimates. Hence we need a modeling approach that encodes the information from the old data compactly in the model itself and furthermore can *integrate* new data into an existing model to refine it.

We propose to use a probabilistic graphical model known as the *factor graph* [4] to represent pathway parameter estimates. We view a factor graph as a representation of a probability function  $p(k_1, k_2, \dots)$  over the parameters  $k_1, k_2, \dots$ . A particular estimate of parameter values has high probability if it fits well with experimental data according to a suitable error measure. A factor graph represents many parameter estimates of varying quality, encoded as a probability function, rather than a single best estimate based on the existing data. A large pathway model typically involves many parameters. As a result,  $p(k_1, k_2, \dots)$  is a high-dimensional function, which is expensive to compute and store. A key advantage of the factor graph model is that it exploits the *network structure* of a pathway to factor  $p(k_1, k_2, \dots)$  as a product of lower-dimensional functions. This drastically reduces the complexity of representing  $p(k_1, k_2, \dots)$  and allows parameter estimates to be refined efficiently.

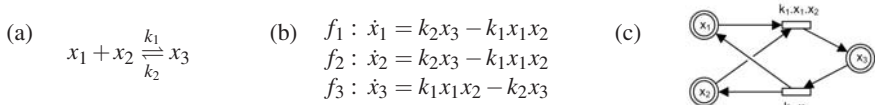
To incorporate new data, we add new nodes to a factor graph and apply a probabilistic inference technique known as *belief propagation* (see [5] for a survey) to refine the parameter estimates represented by  $p(k_1, k_2, \dots)$ . Belief propagation reconciles the local constraints encoded in the new and the old factor graph nodes and ensures that they are globally consistent.

To test our approach, we applied it to the Akt-MAPK pathways. The kinase Akt plays an important role in regulating cellular functions, including, in particular, apoptosis, and has been identified as a major factor in several types of cancer. We created multiple data sets through simulation and introduced them one at a time into the factor graph model. The results show that our approach can obtain estimates that fit the data well and refine them incrementally when new data becomes available.

## 2 Background

### 2.1 Modeling Pathway Dynamics

The dynamics of a signaling pathway is often modeled as a system of nonlinear ordinary differential equations (ODEs):



**Fig. 1.** (a) A reaction, in which two substrates  $x_1$  and  $x_2$  bind reversibly to form a complex  $x_3$ . The speed of the forward and backward reactions depends on the kinetic rate constants  $k_1$  and  $k_2$ , respectively. (b) The corresponding system of ODEs. (c) The HFPN model. The places are drawn as circles, and the transitions, as rectangles.

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), x_2(t), \dots; k_1, k_2, \dots) \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t), \dots; k_1, k_2, \dots), \\ &\vdots \end{aligned} \quad (1)$$

where  $x_i(t)$  denotes the concentration level of molecular species  $i$  at time  $t$  and  $\dot{x}_i(t)$  denotes the corresponding rate of change. Each function  $f_i$ , usually nonlinear, encodes the kinetics of the reactions that produce or consume  $x_i$ . The reactions are typically modeled with the mass action law or Michaelis-Menten kinetics [6], and we assume that the functions  $f_1, f_2, \dots$  are given. The kinetic rate constants  $k_1, k_2, \dots$  are parameter that govern the speed of reactions. See Fig. 1 for an example.

Using the vector notation, we can rewrite (1) more concisely as  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t); \mathbf{k})$ , where  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots)$ ,  $\dot{\mathbf{x}}(t) = (\dot{x}_1(t), \dot{x}_2(t), \dots)$ , and  $\mathbf{k} = (k_1, k_2, \dots)$ . Finally, we also need to specify the initial concentration levels  $\mathbf{x}(0) = \mathbf{x}_0$ .

The system of ODEs in (1) can be represented as a hybrid functional Petri net (HFPN) [7], which makes pathway structure explicit. A HFPN is a directed bipartite graph consisting of two types of nodes: *places* and *transitions*. In our case, places represent molecular species, and transitions represent reactions. The places and transitions are connected by *arcs* to indicate the flow of reactants and products. For an enzyme-catalyzed reaction, a *read arc*, shown pictorially as a dashed arc, connects an enzyme place to a catalyzed transition. It indicates that the enzyme influences, but is not consumed by the reaction. See [7] for more details on the HFPN model.

## 2.2 Parameter Modeling

An important step in building a pathway model is to determine the pathway parameters, which include kinetic rate constants and initial concentration levels of molecular species. Here we mainly deal with unknown kinetic rate constants, but the basic idea applies to unknown initial concentration levels as well.

Experimental determination of parameter values *in vitro* may not be possible or prohibitively expensive. A more practical approach is to estimate the parameter values based on experimental data. Suppose that we are given a set  $D$  of experimental data  $\{\tilde{x}_{ij}\}$ , where  $\tilde{x}_{ij}$  is the experimentally measured concentration level of molecular species  $i$  at time  $T_j$ . The goal is to determine the values of the unknown parameters  $\mathbf{k}$  so that the resulting pathway dynamics, *i.e.*, the evolution of molecular concentration levels over time, fits experimental data well. Mathematically, our goal consists of minimizing an objective function measuring the error in fit to data:

$$J(\mathbf{k}|D) = \sum_{i \in M} \sum_j (x_i(T_j; \mathbf{k}) - \tilde{x}_{ij})^2, \quad (2)$$

where  $M$  denotes the set of experimentally measured molecular species, and  $x_i(t; \mathbf{k}), i = 1, 2, \dots$  are the solution to the system of ODEs in (1) with parameters  $\mathbf{k}$ . Typically we obtain  $x_i(t; \mathbf{k}), i = 1, 2, \dots$  by simulating (1), using a numerical method. We can generalize  $J(\mathbf{k}|D)$  by multiplying each term in (2) by a weight  $w_{ij}$  to favor the fit to data for some species at certain time over others. In the following, however, we use (2) to simplify the presentation. For multiple data sets  $D_1, D_2, \dots, D_n$ , we simply sum up the error due to each data set and denote the total error by  $J(\mathbf{k}|D_1, D_2, \dots, D_n)$ .

Standard estimation algorithms traverse the space of all parameter values and search for an optimal set of values with the best fit with  $D$ . A major challenge is that the size of the parameter space grows exponentially with the number of unknown parameters. Many different search strategies have been proposed to overcome this challenge, including local strategies (such as gradient descent) and global strategies (such as simulated annealing and evolutionary algorithms). See [3] for a survey, as well as [8,9]. However, almost all current algorithms aim to find a single best parameter estimate based on the data available. This is inadequate for incremental pathway modeling: the single estimate cannot be easily improved when new data arrives. We propose instead to use a factor graph to represent a probability distribution that encodes multiple parameter estimates. Using this representation, we can refine the estimates systematically by adjusting their probabilities when new data becomes available.

Yoshida *et al.* adopts a similar probabilistic, data-driven view of parameter estimation [8], but their method assumes that *all* the data is available and is not geared towards incremental modeling. Factor graphs have been used to model biological systems [10], but the main goal there is to study the functional correlations among the molecular species in the pathway rather than the dynamics. An early use of belief propagation in computational biology is to predict protein secondary structure assignment [11].

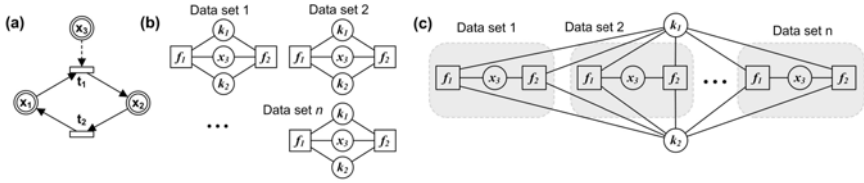
### 3 Incremental Pathway Parameter Modeling

#### 3.1 Overview

Often, experimental data are obtained in an incremental fashion. As a new data set  $D_n$  arrives at some time  $T_n$  with  $T_1 < T_2 < T_3 < \dots$ , we want to incorporate  $D_n$  and compute a new estimate of the parameters  $\mathbf{k}$ . A simplistic approach would be to use all the data available up to time  $T_n$ ,  $\cup_{i=1}^n D_i$ , and recompute the estimate of  $\mathbf{k}$  from scratch. The error in fit to data is then given by  $J(\mathbf{k}|D_1, D_2, \dots, D_n)$ . This approach, however, may be infeasible, because experimental data are generated by different research groups at different times. While the estimated parameter values may be published and accessible, the data used to produce these estimates is usually not. Recomputing the parameter estimate is also inefficient, as it does not take advantage of earlier estimates.

We would like to compute an estimate of  $\mathbf{k}$  at time  $T_n$  using only  $D_n$  and the estimates obtained from the earlier data  $\cup_{i=1}^{n-1} D_i$ . To do so, we encode a set of estimates of  $\mathbf{k}$  as a probability function

$$p(\mathbf{k}|D) = (1/\lambda) \exp(-J(\mathbf{k}|D)), \quad (3)$$



**Fig. 2.** (a) The HFPN model of an enzyme-mediated reversible reaction. (b) A factor graph  $S_n$  is constructed for each data set  $D_n$ . (c) The factor graphs are merged by fusing their common variable nodes representing unknown parameters.

where  $D$  is a given data set,  $\lambda$  is a normalizing constant ensuring that  $\int p(\mathbf{k}|D) d\mathbf{k} = 1$ , and  $J(\mathbf{k}|D)$  measures the error in fit to data, as defined in (2). The probability function  $p(\mathbf{k}|D)$  encodes a set of parameter estimates, with large  $p(\mathbf{k}|D)$  value indicating small error in fit to the data set  $D$ . In other words, we view  $p(\mathbf{k}|D)$  as a probabilistic weight on  $\mathbf{k}$ , expressing preferences over  $\mathbf{k}$  values due to the constraints from the data set  $D$ . Now suppose that  $p(\mathbf{k}|D_1, D_2, \dots, D_{n-1})$  represents the parameter estimates at time  $T_{n-1}$ . When a new data set  $D_n$  arrives at  $T_n$ , we use  $D_n$  to update the probabilistic weights on the estimates encoded by  $p(\mathbf{k}; D_1, D_2, \dots, D_{n-1})$  and obtain a new probability function  $p(\mathbf{k}; D_1, D_2, \dots, D_{n-1}, D_n)$ . This is similar to Bayesian update, except that  $p(\mathbf{k}|D)$  is basically a weight on  $\mathbf{k}$  that depends on the error in fit to data  $J(\mathbf{k}|D)$  and does not in itself have any real statistical meanings.

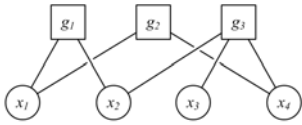
This incremental approach would be beneficial only if we can store and update  $p(\mathbf{k}|D)$  efficiently. For a large pathway model with many unknown parameters,  $p(\mathbf{k}|D)$  is a high-dimensional global function over the entire parameter space. However, each species in a typical signaling pathway interacts with only a small number of other species (see Fig. 5 for an example). We can exploit this insight on the *network structure* of a pathway to approximately factor the high-dimensional function  $p(\mathbf{k}|D)$  into a product of lower-dimensional functions, and represent this factored probability function as a *factor graph* [4]. When combined with belief propagation (see Section 4), this representation helps us to find the best parameter estimates efficiently. Furthermore, it enables us to store and update  $p(\mathbf{k}|D)$  efficiently in an incremental fashion.

Let  $S_n$  be a factor graph representing  $p(\mathbf{k}|D_n)$ , which, as mentioned earlier, represents preferences over  $\mathbf{k}$  values due to the constraints from the data set  $D_n$ . In our incremental approach to parameter modeling, we compute a sequence of factor graphs  $K_n, n = 1, 2, \dots$ , where  $K_1 = S_1$  and  $K_n$  for  $n \geq 2$  is obtained by merging  $S_n$  into  $K_{n-1}$ . See Fig. 2 for an illustration. The merging process uses belief propagation to combine the preferences on  $\mathbf{k}$  values represented by  $K_{n-1}$  with those represented by  $S_n$ . This results in new preferences represented by  $K_n$ .

We are ready to present the factor graph model for  $p(\mathbf{k}|D)$ . We begin with a brief introduction to factor graphs. We then describe how to construct a factor graph  $S_n$ , given a data set  $D_n$  and how to merge  $S_1, S_2, \dots, S_n$  incrementally to build  $K_n$ .

### 3.2 Factor Graphs

Suppose that a high dimensional function  $g(\mathbf{z})$  can be factored as a product of lower dimensional functions:  $g(\mathbf{z}) = \prod_i g_i(\mathbf{z}_i)$ , where  $\mathbf{z} = (z_1, z_2, \dots)$  is a set of variables



**Fig. 3.** The factor model for the function  $g(x_1, x_2, x_3, x_4) = g_1(x_1, x_2) \cdot g_2(x_1, x_4) \cdot g_3(x_2, x_3, x_4)$ .

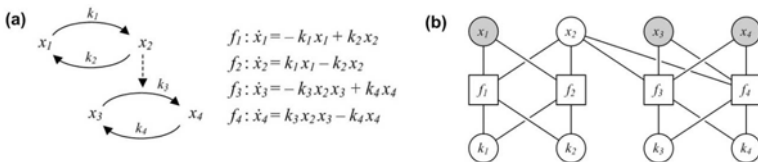
and each  $\mathbf{z}_i$  is a (small) subset of variables in  $\mathbf{z}$ . A factor graph for  $g(\mathbf{z})$  is an undirected bipartite graph consisting of two types of nodes: *factor nodes* and *variable nodes*. Each factor  $g_i(\mathbf{z}_i)$  has a corresponding factor node in  $G$ , and each variable  $z_j$  has a corresponding variable node in  $G$ . There is an undirected edge between the factor node for  $g_i(\mathbf{z}_i)$  and the variable node for  $z_j$  if  $z_j \in \mathbf{z}_i$ , i.e.,  $z_j$  is a variable of the function  $g_i(\mathbf{z}_i)$ . An example is shown in Fig. 3

A variable node for  $z_j$  contains a probability distribution over the values of  $z_j$ . A factor node for  $g_i(\mathbf{z}_i)$  specifies the dependencies among the variables in  $\mathbf{z}_i$  and expresses preferences over their values due to some constraints. In pathway parameter modeling, the main variables are the parameters, and the constraints arise from the ODEs in which a parameter appears. For example, consider the reaction shown in Fig. 1. Suppose that data are available for  $x_1(t), x_2(t), x_3(t)$  at all times  $t$ , but the rate constants  $k_1$  and  $k_2$  are unknown. Then, each of the three equations in the system of ODEs for the reactions imposes a constraint on the unknowns  $k_1$  and  $k_2$  at all times  $t$ . Those combinations of  $k_1$  and  $k_2$  values that satisfy the constraints are favored. In general, each equation in an ODE model represents a local constraint on the parameters involved in the equation, and each such constraint results in a factor node. The resulting factor graph represents the probability function  $p(\mathbf{k}|D)$  as a product of factors, each involving only a small number of unknown parameters.

### 3.3 The Factor Graph Structure

Given a data set  $D$ , we now construct the factor graph  $S$  for the parameters of a system of ODEs modeling a biological pathway. For each equation  $\dot{x}_i = f_i(\mathbf{x}; \mathbf{k})$  in (1), we create a factor node  $v(f_i)$  in  $S$ . We also create a variable node  $v(k_j)$  for each parameter  $k_j$  and a variable node  $v(x_j)$  for each molecular concentration level  $x_j$ . We insert an edge that connects a factor node for  $f_i$  and a variable node for  $k_j$  (or  $x_j$ ), if  $k_j$  (or  $x_j$ ) is involved in  $f_i$ . An example is shown in Fig. 4

Our main goal is to capture the dependencies among the parameters. We can eliminate many of the variable nodes representing molecular concentration levels and thus simplify  $S$ . However, we can eliminate a variable node only if it does not represent the concentration level of an enzyme. The reason is that although enzymes are not consumed in catalytic reactions, their concentration levels influence the reactions. In



**Fig. 4.** (a) A simple signaling cascade and its ODEs. (b) The factor graph representation. The variable nodes in gray— $x_1, x_3$ , and  $x_4$ —can be eliminated.



general, eliminating a variable node corresponding to an enzyme results in the loss of dependency between the reaction producing the enzyme and the reaction catalyzed by the enzyme. To see this, consider again the example in Fig. 4. If we eliminate the variable nodes for  $x_1$ ,  $x_3$  and  $x_4$ , which are not enzymes, the dependencies among  $k_1, k_2, k_3$ , and  $k_4$  remain intact. However, if we eliminate the variable node  $x_2$ , an enzyme, the factor graph breaks into two disconnected components. There is no constraint that connects  $k_1$  and  $k_2$  with  $k_3$  and  $k_4$ , implying that  $k_1$  and  $k_2$  are independent of  $k_3$  and  $k_4$ . This is clearly not the case.

To summarize, the structure of a factor graph—the variable nodes, the factor nodes, and the edges—is constructed from the ODEs that model a signaling pathway. Each factor captures the dependencies among the parameters involved in a particular equation.

### 3.4 The Compatibility Functions

To complete the construction of the factor graph  $S$ , we need to associate a factor, also called a *compatibility function*, with each factor node  $v(f_i)$  and decomposes  $p(\mathbf{k}|D)$  as a product of these compatibility functions. Although all compatibility functions depend on  $D$ , we drop the explicit mention of  $D$  in this section to simplify the notation. It is understood that compatibility functions are defined with respect to a given data set  $D$ . The compatibility function for  $v(f_i)$  is given by

$$g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \exp(-E_i(\mathbf{k}_i, \mathbf{x}_i(t))), \tag{4}$$

where  $\mathbf{k}_i$  and  $\mathbf{x}_i(t)$  are respectively the set of parameters and the set of molecular concentration levels corresponding to the variables nodes connected to  $v(f_i)$ . Note the distinction between  $x_i$ , which denotes the concentration level of species  $i$ , and  $\mathbf{x}_i$ . The function  $E_i(\mathbf{k}_i, \mathbf{x}_i(t))$  consists of two terms:

$$E_i(\mathbf{k}_i, \mathbf{x}_i(t)) = E_{i,1}(\mathbf{k}_i) + E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)). \tag{5}$$

The first term  $E_{i,1}(\mathbf{k}_i)$  measures the fit to data for a particular choice of values for the parameters in  $\mathbf{k}_i$ . The second term  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$  measures whether the values for  $\mathbf{k}_i$  are consistent with those for  $\mathbf{x}_i(t)$ .

We calculate  $E_{i,1}(\mathbf{k}_i)$  based on the global effect of  $\mathbf{k}_i$  on the fit to data for the molecular species that are experimentally measured:

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k} \setminus \mathbf{k}_i} \sum_{m \in M} \sum_j (x_m(T_j; \mathbf{k}) - \tilde{x}_{mj})^2, \tag{6}$$

where  $\mathbf{k} \setminus \mathbf{k}_i$  denotes the set of parameters in  $\mathbf{k}$ , but not in  $\mathbf{k}_i$ ,  $M$  denotes the set of all species that are measured experimentally,  $x_m(t; \mathbf{k})$  is the concentration level of species  $m$  at time  $t$ , obtained by simulating the system of ODEs in (1) with parameters  $\mathbf{k}$ , and finally  $\tilde{x}_{mj}$  is the experimental concentration level of species  $m$  at time  $T_j$ .

The second term  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$  measures the consistency between the parameter values  $\mathbf{k}_i$  and concentration levels  $\mathbf{x}_i(t)$ :  $\mathbf{k}_i$  and  $\mathbf{x}_i(t)$  are *consistent* if  $\mathbf{x}_i(t)$  can be obtained by simulating the system of ODEs in (1) with parameter values  $\mathbf{k}_i$  and some suitable choice of values for parameters in  $\mathbf{k} \setminus \mathbf{k}_i$ . The function  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$  takes binary values. If  $\mathbf{k}_i$  and  $\mathbf{x}_i(t)$  are consistent,  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)) = 0$ ; otherwise,  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)) = +\infty$ . This

way,  $\mathbf{k}_i$  values that are inconsistent with the dynamics defined by the ODEs are filtered out, regardless of their agreement with experimental data according to  $E_{i,1}(\mathbf{k}_i)$ .

With our definition of compatibility functions, the factor graph  $S$  encodes exactly the function

$$g(\mathbf{k}, \mathbf{x}(t)) = \frac{1}{\lambda} \prod_i g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \frac{1}{\lambda} \exp\left(-\sum_i E_i(\mathbf{k}_i, \mathbf{x}_i(t))\right), \quad (7)$$

where  $\mathbf{k} = \cup_i \mathbf{k}_i$ ,  $\mathbf{x} = \cup_i \mathbf{x}_i$ , and  $\lambda$  is a normalizing constant ensuring that  $g(\mathbf{k}, \mathbf{x}(t))$  represents a well-defined probability function. The function  $g(\mathbf{k}, \mathbf{x}(t))$  has the same extremal values as  $J(\mathbf{k})$  and  $p(\mathbf{k})$ :

**Theorem 1.** *The following statements are equivalent:*

1. *The parameter values  $\mathbf{k}^*$  minimize  $J(\mathbf{k})$ .*
2. *The parameter values  $\mathbf{k}^*$  maximize  $p(\mathbf{k})$ .*
3. *The parameter values  $\mathbf{k}^*$  and concentration levels  $\mathbf{x}(t; \mathbf{k}^*)$  maximize  $g(\mathbf{k}, \mathbf{x}(t))$ , where  $\mathbf{x}(t; \mathbf{k}^*)$  is the molecular concentration levels obtained by simulating the ODE model in (1) with parameter values  $\mathbf{k}^*$ .*

The proof is given in Appendix A. This result implies that to minimize  $J(\mathbf{k})$  or maximize  $p(\mathbf{k})$ , we may equivalently maximize  $g(\mathbf{k}, \mathbf{x}(t))$ . Why do we want to do so? The reason is that although  $g(\mathbf{k}, \mathbf{x}(t))$  is also a high-dimensional function, it is factored as a product of lower-dimensional functions represented by the factor graph  $S$ . We can maximize it effectively using belief propagation (Section 4), when searching for a parameter estimate with the best fit to data.

The compatibility functions defined above measure the fit to data globally over all experimentally measured molecular species. As a heuristic for improving efficiency, we introduce a variant which measures the fit to data locally as well. The definition of  $E_{i,1}(\mathbf{k}_i)$  then depends on whether the concentration level  $x_i$  of molecular species  $i$  is measured experimentally. If it is, we calculate  $E_{i,1}(\mathbf{k}_i)$  locally using only the data for  $x_i$ :

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k}|\mathbf{k}_i} \sum_j (x_i(T_j; \mathbf{k}) - \tilde{x}_{ij})^2. \quad (8)$$

If  $x_i$  is not measured experimentally, we calculate  $E_{i,1}(\mathbf{k}_i)$  globally using (6). Intuitively, calculating the fit to data locally strengthens the local constraints and makes belief propagation (Section 4) more greedy. This turns out to be helpful in our experiments (Section 4). However, it does not have the theoretical guarantee stated in Theorem 1.

We now discuss how to represent and compute the compatibility functions  $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$ . First, the parameter values and the concentration levels are discretized into a finite set of intervals. Both the probability distributions for variable nodes and the compatibility functions for factor nodes are represented using this discretization. This is common practice for factor graphs used in conjunction with belief propagation (5). It is not a severe limitation here, as the experimentally measured concentration levels for proteins in a signaling pathway often have very limited accuracy. Furthermore, once belief propagation gives the best parameter estimate up to the resolution of the discretization, we can further refine the estimate by performing a local search, thus mitigating the effect of discretization. More details regarding this can be found in Section 5. One advantage of the discrete representation is that the resulting factor graph can represent

arbitrary probability distributions, up to the resolution of the discretization. There is no need to assume a particular parametric form of the distribution.

Next, to compute  $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$ , we need to perform the minimization required in (6) or (8). For this, we sample a representative set of parameter values and perform the minimization over the set of sampled values. This would be expensive computationally if performed on the space of all parameters. We need sophisticated sampling methods such as Latin square sampling [12] to reduce the computational cost. Whenever possible, we also decompose a pathway model into components (Section 3.5). Sampling is performed only within a pathway component, which usually contains a small subset of parameters. This keeps the computational cost low.

### 3.5 Pathway Decomposition

For computational efficiency, we decompose a pathway into components. Each component usually contains only a small subset of unknown parameters. We build a factor graph  $S'$  for each component independently, assuming that the component is unaffected by the other components. Each component factor graph  $S'$  encodes a probability function expressing preferences over the values of the parameters contained in  $S'$ . To account for the dependency among the parameters from different components, we merge the component factor graphs and apply belief propagation (Section 4) to reconcile the different preferences over parameter values from each component. We do not have space here to describe this somewhat elaborate procedure. The details can be found in [13]. See Fig. 5 for an example of a decomposed pathway model.

### 3.6 Data Integration

Suppose that a sequence of data sets  $D_1, D_2, \dots$  arriving at time  $T_1, T_2, \dots$ . Let  $K_n$  denote the factor graph for  $p(\mathbf{k}|D_1, D_2, \dots, D_n)$ . We want to build  $K_n$  incrementally by integrating the data sets one at a time. At the  $n$ th stage, we first apply the procedure described above to construct a factor graph  $S_n$  for  $D_n$ . To construct  $K_n$ , we merge  $S_n$  with  $K_{n-1}$  by fusing their common variable nodes. Specifically, if a node of  $S_n$  represents the same unknown parameter as a node of  $K_{n-1}$ , they are merged as a single node in  $K_n$ . The edges are rearranged accordingly. Other nodes of  $S_n$  and  $K_{n-1}$  remain the same in  $K_n$ . See Fig. 2 for an illustration. It is important to note that although  $K_n$  takes into account all the data  $\bigcup_{i=1}^n D_i$ , the construction of  $K_n$  requires only  $D_n$ . Information from the earlier data sets  $\bigcup_{i=1}^{n-1} D_i$  is encoded in  $K_{n-1}$ . Intuitively each new data set  $D_n$  adds a “slice” to our final factor graph  $K_n$ . So the size of  $K_n$  grows linearly with  $n$ .

We now turn to the important step of belief propagation, which reconciles the local constraints encoded by  $K_{n-1}$  and  $S_n$ .

## 4 Finding the Best Parameter Estimate

Theorem 1 shows that to find the minimum  $\mathbf{k}^*$  of  $J(\mathbf{k}|D_1, D_2, \dots, D_n)$ , we can equivalently maximize  $g(\mathbf{k}, \mathbf{x}(t))$  represented by the factor graph  $K_n$ . We compute the maximum by applying a standard belief propagation (BP) algorithm called the max-product algorithm to  $K_n$ .

We give only a quick overview of BP here. See [4,5] for comprehensive tutorials. Let  $G$  be a factor graph representing a factored non-negative function  $g(\mathbf{z}) = g(z_1, z_2, \dots) = \prod_i g_i(\mathbf{z}_i)$ , where  $\mathbf{z}_i$  is the subset of variables involved in the factor  $g_i(\mathbf{z}_i)$ . After normalization,  $g(\mathbf{z})$  can be considered a probability function. Each variable node  $v(z_j)$  of  $G$  is initialized with a probability distribution  $\pi_0(z_j)$ —commonly called a *belief*—over the values of  $z_j$ . A preferred  $z_j$  value has higher probability. The initial distribution  $\pi_0(z_j)$  represents our prior knowledge on the value of  $z_j$ . If there is no prior information on  $z_j$ , we set its initial distribution to be uniform. After initialization, a variable node  $v(z_j)$  sends its belief  $\pi(z_j)$  as a message to each adjacent factor node  $v(g_i)$ . Upon receiving the messages from the adjacent variable nodes, a factor node  $v(g_i)$  combines them with its own compatibility function  $g_i(\mathbf{z}_i)$  and creates a new message, which is sent to each variable node  $v(z_j)$  adjacent to  $v(g_i)$ . The belief at  $v(z_j)$  is then updated so that  $z_j$  values satisfying the compatibility function  $g_i(\mathbf{z}_i)$  well have their probabilities increased. The order in which to send the messages must follow a suitable protocol, and the messages stop when a termination condition is met.

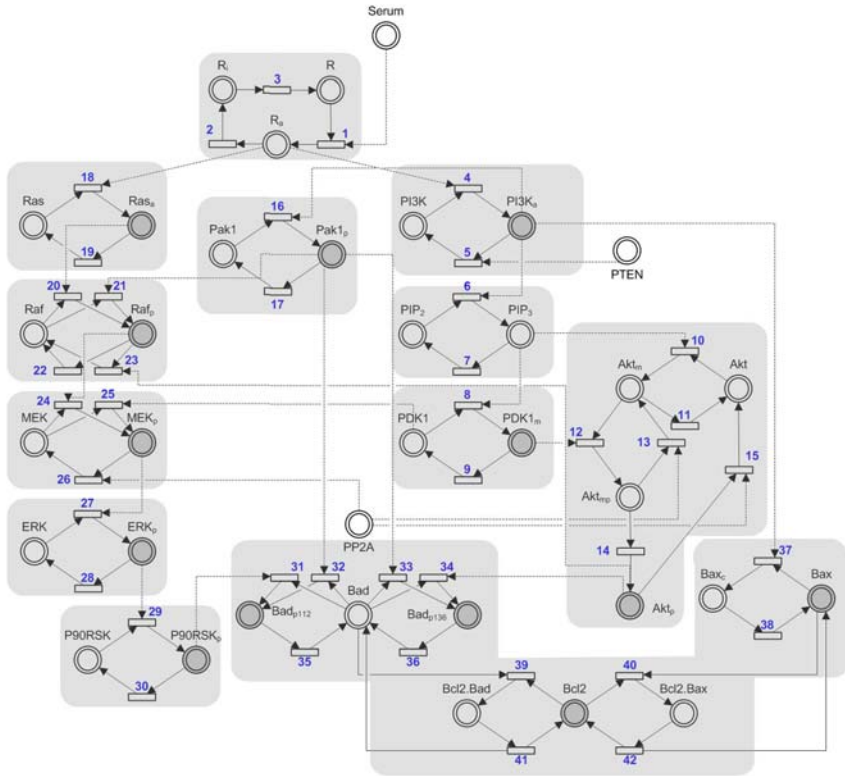
When BP terminates, the variable nodes take on beliefs favoring values that satisfy well the local constraints represented by the compatibility functions in the factor nodes. If a factor graph  $G$  contains no cycles, BP converges to the *global* maximum of the function that  $G$  represents [14]. In practice, a factor graph modeling a complex system often contains cycles. So convergence is not guaranteed, and one needs to terminate the algorithm using heuristic criteria. Nevertheless, BP on general factor graphs has generated good results in diverse applications [15,16]. One reason is that BP is in essence a dynamic programming algorithm, which performs a more global search than strategies such as gradient descent, and is less likely to get stuck in local maxima.

We apply BP to a factor graph representing the function  $g(\mathbf{k}, \mathbf{x}(t))$  in (7). Each compatibility function  $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$  in the factor graph encodes two types of constraints:  $E_{i,1}(\mathbf{k}_i)$  measures the fit to data, and  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$  measures the consistency between  $\mathbf{k}_i$  and  $\mathbf{x}_i(t)$  with respect to the dynamics defined by the ODEs in (1). BP favors  $\mathbf{k}$  and  $\mathbf{x}$  values that satisfy these constraints well. It is also important to remember that when BP terminates, the variable nodes of the factor graph contain not only the parameter values with the best fit to existing data, but also alternative parameter values of varying quality weighted by the probabilities. These alternatives will become useful when new data arrives.

We run BP on each incrementally constructed factor graph  $K_n$ . For  $n = 1$ , the variable nodes of  $K_1$  are initialized with the uniform probability distribution. For  $n \geq 2$ , the variable nodes of  $K_n$  are initialized with beliefs resulting from BP at the previous stage. Recall that  $K_n$  is obtained by merging  $K_{n-1}$  with a factor graph slice  $S_n$  representing the new data set  $D_n$  (Section 3.6). So BP has the effect of reconciling the constraints due to the new data (encoded in  $S_n$ ) with those due to the earlier data (encoded in  $K_{n-1}$ ) and favoring those parameter values with good fit to both the new and the old data.

## 5 Results

We tested our approach on the Akt-MAPK signaling pathways. The kinase Akt is a major factor in many types of cancer. The Akt pathway is one of the most actively



**Fig. 5.** The HFPN model of the Akt-MAPK pathways. A place node in the model is shaded in gray if data is available for the corresponding molecular species. The light gray boxes indicate the components obtained through pathway decomposition.

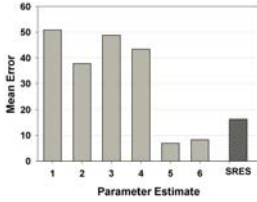
studied kinase pathways, as it plays a key role in essential cellular functions, including cell survival, differentiation, *etc.* [17]. The Akt pathway interacts with several other pathways while performing its functions, in particular, the MAPK pathway.

In our earlier work [18], we performed parameter estimation on a combined model of the Akt-MAPK pathways using experimental data and studied the crosstalk between them. In the present setting, due to the lack of sufficient number of experimental data sets, we used synthetic data. The Akt-MAPK model used in our case study contains 36 molecular species and 42 unknown parameters. See Fig. 5. A larger figure along with model parameter values is available at <http://www.comp.nus.edu.sg/~rpsysbio/recomb2010>. We generated six data sets by simulating the model under different knockdown conditions, in which the initial concentration level of each of six molecular species—Akt, PDK1, PP2A, PTEN, and the cell receptor—is reduced. Each data set contains concentration levels of 13 molecular species at 50 time points.

We normalized the value of each parameter to a range between 0 and 1 and divided the range into 10 equally-sized intervals. Due to the discretization, belief propagation produces the best parameter intervals rather than exact values. As a post-processing

step, we apply the Levenberg-Marquardt algorithm [19], starting from the mid-points of the best parameter intervals obtained from belief propagation. This gives us the final parameter estimate that minimizes the error in fit to data.

As mentioned in Section 1, a key goal of our work is to address the issue of not having access to all the data at the same time. In our test, we introduced the six data sets one at a time, in an arbitrary, but fixed order. At each stage, to perform parameter estimation, we used only one data set along the factor graph model from the earlier estimation; all other data sets were kept away. Since sampling is used during the factor graph modeling (Section 3.4), we repeated each test 10 times.



**Fig. 6.** The error in fit to data, as six data sets were introduced one at a time. The darker bar indicates the error of the parameter estimate obtained by SRES, using all six data sets.

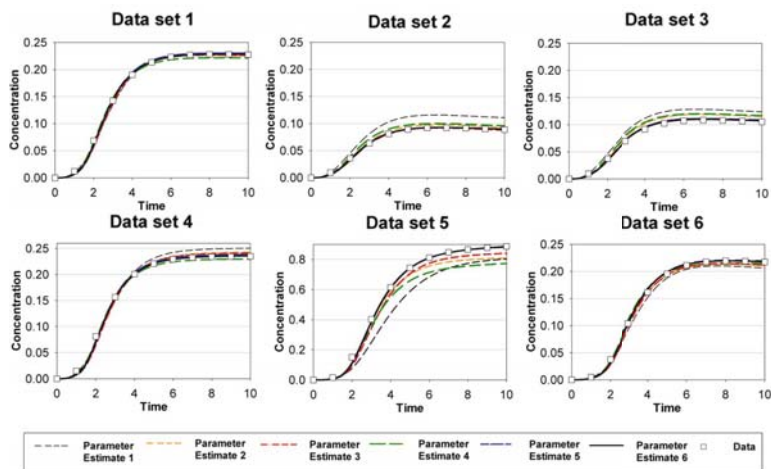
Fig. 6 shows the mean error in fit to data over the 10 runs in each stage. To examine the benefits of using multiple data sets for parameter estimation, the error is measured according to (2) using all six data sets. The plot shows that as more data are used, the error generally decreases, as expected. Fig. 7 shows the concentration level of Bad (Bcl2 antagonist of cell death), an important downstream protein in the pathway. Each plot shows how the concentration level changes over time under one of the six knockdown conditions. Figs. 6 and 7 indicate that the fit to data improves, as more data sets are introduced to refine the parameter estimate. For example, parameter estimate 1 causes substantial error in fit to data set 2, 3, and 5, while parameter estimate 6, after integrating all data, fits well with all data sets. The results confirm that our approach can integrate new data and improve the parameter estimates effectively.

Next, we compared our results with that from COPASI [20], a well-established pathway simulator with parameter estimation functions. COPASI contains several methods for parameter estimation. We used SRES, which is the best based our experiences. We ran SRES for an extended duration (10 hours), using all *six data sets*. After integrating enough data sets, our approach of incremental parameter modeling obtained comparable and better estimates (Fig. 6). The results suggest that our incremental approach through data integration does not sacrifice parameter estimation accuracy, compared with global estimation methods that require access to all the data sets at once.

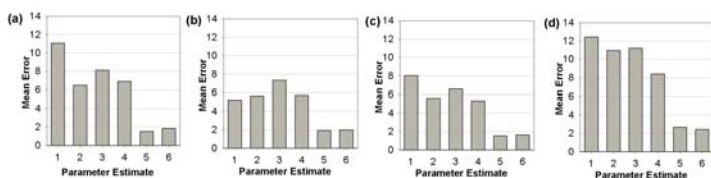
To test the robustness of our approach, we considered four additional knockdown conditions by combining the knockdown conditions specified earlier. We generated four new data sets under these additional conditions and computed the error in fit to data for the six parameter estimates obtained earlier (Fig. 8). We did not recompute the parameter estimates using the additional data, as the purpose here is to check the robustness of the estimates obtained earlier under new conditions. The results indicate a trend similar to that shown in Fig. 6.

As more data sets are integrated, we expect that the uncertainty of parameter estimates decreases. Fig. 9 shows the change in the standard deviations of some estimated parameters as the number of data sets increases. There is a general decrease in the standard deviations for all estimated parameters, indicating that data integration is effective

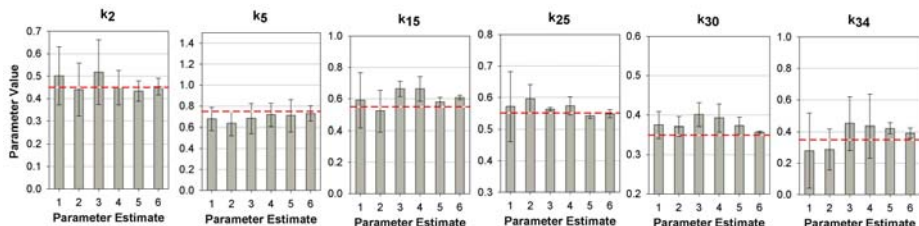




**Fig. 7.** The change in concentration level over time for  $\text{Bad}_{p136}$  under six knockdown conditions. The six curves in each plot correspond to the six different parameter estimates as more data sets are integrated. Data points are shown every 5 time steps to avoid cluttering the plots.



**Fig. 8.** The error in fit to data under combination of knockdown conditions



**Fig. 9.** The mean values and the standard deviations of the estimated parameters over 10 runs. The bars indicate the mean values of estimated parameters. The error bars indicate the standard deviations. The dashed lines indicate the nominal parameter values.

for reducing the uncertainty of estimated parameters. For some parameters, such as the ones shown in Fig. 9, the uncertainty is very low, after six data sets were integrated. However, for some other parameters, including  $k_7, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{21}, k_{22}, k_{32}, k_{37}, k_{40}, k_{41}, k_{42}$ , the uncertainty remains large. The graphical model of the pathway (Fig. 5) reveals that such parameters are mostly associated with molecular species that are either

(i) involved in several reactions, *e.g.*, Akt<sub>m</sub>, Raf, Bad, or (ii) have insufficient data to constrain their values, *e.g.*, PIP<sub>3</sub>, Akt<sub>m</sub>. This observation suggests that biological pathways are less sensitive to parameter variations around molecular species involved in more than one set of production-consumption reactions. The uncertainty level in parameter estimates can also provide guidance to biologists in the subsequent design of their experiments to further constrain important pathway parameters.

## 6 Conclusion

Pathway model construction is often an incremental process, as new experiments lead to discoveries of additional players and interactions in a pathway. This paper presents a data integration approach to incremental pathway parameter modeling. We use the factor graph as a probabilistic model of pathway parameter estimates. It enables us to refine the parameter estimates in a principled and efficient manner, when new data becomes available. A main benefit of our approach is that the factor graph model compactly encodes the information from old data in itself and uses only new data to refine the parameter estimates. It eliminates the unrealistic requirement of having access to all data, both old and new, in order to improve the parameter estimates.

Several aspects of our approach require further investigation. So far, we have only tested it with unknown kinetic rate constants as parameters. Our approach can also deal with unknown initial molecular concentration levels by treating them as parameters, but we are yet to implement and test our approach to handle this variant. We also need to test this method on multiple signaling pathway models using real experimental data.

An important underlying idea of our approach is to *compose* factor graph models. The current work exploits temporal composition by merging successive slices of factor graphs representing new data sets. This allows us to integrate new data and refine model parameters. We can go one and exploit spatial composition. When new experiments suggest additional components of a pathway or interacting pathways, we may compose the models for these components and pathways to form a single model. Spatial composition allows us to expand a model and incorporate missing players and interactions. The pathway decomposition technique described briefly in Section 3.5 in fact constitutes a special case of spatial composition, but more work is needed to explore spatial composition methods. Together temporal and spatial compositions create a modeling framework that supports model refinement and expansion systematically.

*Acknowledgments.* We thank Lisa Tucker-Kellogg for many fruitful discussions. This work is supported in part by AcRF grant R-252-000-350-112 from the Ministry of Education, Singapore.

## References

1. Bhalla, U.S., Iyengar, R.: Emergent properties of networks of biological signaling pathways. *Science* 283, 381–387 (1999)
2. Aldridge, B.B., Burke, J.M., Lauffenburger, D.A., Sorger, P.K.: Physicochemical modelling of cell signalling pathways. *Nature Cell Biology* 8(11), 1195–1203 (2006)



3. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research* 13(11), 2467–2474 (2003)
4. Kschischang, F., Frey, B., Loeliger, H.: Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory* 42(2), 498–519 (2001)
5. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge (2009)
6. Klipp, E., et al.: *Systems Biology in Practice*. Wiley-VCH, Chichester (2005)
7. Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., Miyano, S.: Biopathways representation and simulation on hybrid functional Petri net. *Silico Biology* 3(3), 389–404 (2003)
8. Yoshida, R., Nagasaki, M., Yamaguchi, R., Imoto, S., Miyano, S., Higuchi, T.: Bayesian learning of biological pathways on genomic data assimilation. *Bioinformatics* 24(22), 2592–2601 (2008)
9. Purvis, J., Radhakrishnan, R., Diamond, S.: Steady-state kinetic modeling constrains cellular resting states and dynamic behavior. *PLoS Computational Biology* 5(3) (2009)
10. Gat-Viks, I., Tanay, A., Raijman, D., Shamir, R.: The factor graph network model for biological systems. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) *RECOMB 2005*. LNCS (LNBI), vol. 3500, pp. 31–47. Springer, Heidelberg (2005)
11. Delcher, A., Kasif, S., Goldberg, H., Hsu, W.: Protein secondary structure modelling with probabilistic networks. In: *Proc. Int. Conf. on Intelligent Systems & Molecular Biology*, pp. 109–117 (1993)
12. Kalos, M., Whitlock, P.: *Monte Carlo Methods*, vol. 1. John Wiley & Sons, New York (1986)
13. Koh, G.: *Pathway Models Decomposition and Composition Techniques for Parameter Estimation*. PhD thesis, Graduate School of Integrative Sciences, National University of Singapore, Singapore (2008)
14. Pearl, J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco (1988)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. *Int. J. Computer Vision* 70(1), 41–54 (2004)
16. McEliece, R.J., MacKay, D.J., Cheng, J.F.: Turbo decoding as an instance of Pearl’s “Belief Propagation” algorithm. *IEEE J. on Selected Areas in Communications* 16(2), 140–152 (1998)
17. Brazil, D.P., Yang, Z.Z., Hemmings, B.A.: Advances in protein kinase B signalling: AKTion on multiple fronts. *Trends in Biochemical Sciences* 29(5), 233–242 (2004)
18. Koh, G., Teong, H.F.C., Clément, M.V., Hsu, D., Thiagarajan, P.: A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. *Bioinformatics* 22(14), e271–e280 (2006)
19. Gill, P., Murray, W., Wright, M.: *Practical Optimization*. Academic Press, London (1982)
20. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI—a Complex Pathway Simulator. *Bioinformatics* 22(24), 3067–3074 (2006)

### A Proof of Theorem 1

*Proof.* Since  $p(\mathbf{k}) = (1/\lambda) \exp(-J(\mathbf{k}))$  and the exponential function is monotonic, the equivalence between statements 1 and 2 clearly holds.

We now prove the equivalence between statements 1 and 3. Define  $E(\mathbf{k}, \mathbf{x}(t)) = \sum_i E_i(\mathbf{k}_i, \mathbf{x}_i(t))$ . Since we want to minimize  $E(\mathbf{k}, \mathbf{x}(t))$ , we are only interested in the case when  $E(\mathbf{k}, \mathbf{x}(t))$  is finite. The function  $E(\mathbf{k}, \mathbf{x}(t))$  is finite if and only if  $\mathbf{k}_i$  and  $\mathbf{x}_i$  are consistent for all  $i$ . Let  $\mathbf{x}(t; \mathbf{k})$  denote the concentration levels consistent with the parameters  $\mathbf{k}$ . In this case,  $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)) = 0$  for all  $i$ . Using this and (6), we then get

$$\begin{aligned} \min_{\mathbf{k}} E(\mathbf{k}, \mathbf{x}(t; \mathbf{k})) &= \min_{\mathbf{k}} \left( \sum_i E_{i,1}(\mathbf{k}_i, \mathbf{x}_i(t)) \right) \\ &= \min_{\mathbf{k}} \left( \sum_i \min_{\mathbf{k} \setminus \mathbf{k}_i} \sum_{m \in M} \sum_j (x_m(t_j; \mathbf{k}) - \tilde{x}_{mj})^2 \right) \\ &= \min_{\mathbf{k}} \left( \sum_i \min_{\mathbf{k} \setminus \mathbf{k}_i} J(\mathbf{k}) \right) \end{aligned} \tag{9}$$

Note that  $\min_{\mathbf{k} \setminus \mathbf{k}_i} J(\mathbf{k})$  is a function of  $\mathbf{k}_i$ . If  $\mathbf{k}^*$  minimizes  $J(\mathbf{k})$ , then  $\mathbf{k}_i^*$  minimizes  $\min_{\mathbf{k} \setminus \mathbf{k}_i} J(\mathbf{k})$  for all  $i$ . It then follows from (9) that

$$\min_{\mathbf{k}} E(\mathbf{k}, \mathbf{x}(t; \mathbf{k})) = \sum_i \min_{\mathbf{k}} J(\mathbf{k}).$$

Since  $g(\mathbf{k}, \mathbf{x}(t)) = (1/\lambda) \exp(-E(\mathbf{k}, \mathbf{x}(t)))$ , the conclusion follows. □

# The Poisson Margin Test for Normalisation Free Significance Analysis of NGS Data

Adam Kowalczyk<sup>1,2,\*</sup>, Justin Bedo<sup>1,2</sup>, Thomas Conway<sup>1,3</sup>,  
and Bryan Beresford-Smith<sup>1,2</sup>

<sup>1</sup> NICTA, Victoria Research Laboratory

<sup>2</sup> Department of Electrical and Electronic Engineering

Adam.Kowalczyk@nicta.com.au

<sup>3</sup> Department of Computer Science and Software Engineering  
The University of Melbourne, Parkville, VIC 3010, Australia

**Abstract. Motivation:** The current methods for the determination of the statistical significance of peaks and regions in NGS data require an explicit normalisation step to compensate for (global or local) imbalances in the sizes of sequenced and mapped libraries. There are no canonical methods for performing such compensations, hence a number of different procedures serving this goal in different ways can be found in the literature. Unfortunately, the normalisation has a significant impact on the final results. Different methods yield very different numbers of detected “significant peaks” even in the simplest scenario of ChIP-Seq experiments which compare the enrichment in a single sample relative to a matching control. This becomes an even more acute issue in the more general case of the comparison of multiple samples, where a number of arbitrary design choices will be required in the data analysis stage, each option resulting in possibly (significantly) different outcomes.

**Results:** In this paper we investigate a principled statistical procedure which eliminates the need for a normalisation step. We outline its basic properties, in particular the scaling upon depth of sequencing. For the sake of illustration and comparison we report the results of re-analysing a ChIP-Seq experiment for transcription factor binding site detection. In order to quantify the differences between outcomes we use a novel method based on the accuracy of *in silico* prediction by SVM-models trained on part of the genome and tested on the remainder.

**Availability:** The supplementary material is available at [\[1\]](#).

## 1 Introduction

Current short read sequencing technology (routinely referred to as Next Generation Sequencing or *NGS*) allows for genome wide scans for various phenomena of interest, such as methylation, transcription factor binding sites, etc. In order to derive meaningful results from the mapping of short reads (or tags) to the

---

\* Corresponding author.

reference genome, a number of statistical filters based on the binomial distribution, Poisson distribution and their variants have been proposed in the literature [2,3,4,5]. These methods are also very similar to SAGE data analysis [6,7], which also deals with short sequence data.

In order to discern a meaningful signal from tags mapped to a reference genome, a number of biases have to be dealt with and corrected for, as the signal “is actually the convolution of a number of effects: the density of mappable bases in a region, the underlying chromatin structure and the actual signal from transcription factor binding” [2]. The natural way to mitigate these issues is to introduce control samples, so that the detected signal is in the form of local enrichment of tag counts with respect to the control. The closer the preparation and processing of the control to the target sample, the more reliable the mitigation of biases.

However, experimenters cannot ensure that the target and control samples are prepared and processed in a completely equivalent manner and in practice the number of tags derived from two separate sequencing reaction can differ by a significant factor. This situation becomes endemic if one attempts to develop local models [2] compensating for local biases along the reference DNA. In the simplest situation such as ChIP-Seq experimental detection of binding sites for a transcription factor [2], where one attempts to detect enrichments in the target sample with respect to a carefully prepared control, there is a plausible argument for scaling the control sample counts to the level of the target, especially when considering pre-filtered narrow regions of significant enrichment.

In practice, there are other scenarios where such scaling is less applicable. An example is where one looks for the differential peaks between two different tissue samples, e.g. differences in methylation between two cell lineages [8,5]. Here, even the direction of scaling (local or global) is not obvious. Moreover, as argued in [5], the common level to which the sample counts are adjusted has a profound impact on the statistical significance of peaks when either Poisson or binomial models are used (see Section 3.3), thus the number of detected peaks depends significantly on the choices of scaling strategies.

The situation becomes even more cumbersome for experiments that involve multiple samples, for example when quantifying the difference between two cell lineages using pairs of samples collected from multiple subjects in order to account for patient specific heterogeneity. One possibility is to adjust all counts to a common size across the whole collection. As we have noted, this common size impacts on the number of “significant peaks” as in practice scaling could be by a factor of 2 or more with the resulting variation in  $p$ -values being by a factor of 4 or more. Moreover, the addition of new samples to the analysis could lead logically to readjustment of the updated “common size” distorting the previous results. The ad hoc nature of some of these adjustments undermines the principled statistical analysis, introducing arbitrary design choices and obscure data driven adjustments.

In this paper we propose a different statistical technique that does not require an explicit sample size adjustment and thus functions directly on the original counts. Any adjustments can be used as an additional means for accounting

for other biases in the data. An example of this is the known variations in the density of mappable tags (i.e., the effective depth of sequencing) [2] along the DNA sequence.

## 2 Background

We now outline a conceptual model which can be used while reading this paper. Using a specific protocol (sonication, enzymatic reactions cutting the DNA at specific locations, protein immunoprecipitation selecting specific fragments of protein bound to them, etc.) a library  $L$  of DNA fragments is prepared. From this library a subset  $R$  is randomly sampled and for each of the sampled fragments a part of it, a *short read*, is sequenced providing a *tag*, which is a  $k$ -mer of DNA bases. The tag is then mapped to one or many matching locations in an a priori known reference DNA sequence, the human genome in our case, using a specific protocol, e.g. only exact matching, or only exact and unique, or only unique with up to one error, etc. We are interested in the reference genome locations where significant over/under-representation of the mapped tags occur, so called *peaks* or *peak ranges*, as these can be interpreted as evidence for some specific property of DNA or its epigenetic modifications. In some experiments such as SAGE-Seq or Digital Karyotyping [8,5] there are natural peak regions, as the tags congregate at specific DNA locations determined by the enzymes used to cut the source DNA. In other cases, such as ChIP-Seq experiments using sonication, the peak regions have to be determined from the data using specific algorithms (e.g. [2,9]), or perhaps just defined by a simple partitioning of the genome into uniform small blocks, say of the order of a thousand bases.

In this section we assume that a set of peak regions of interest is given to us. Let us consider a single peak range  $r$ . We denote by  $X = X(R)$  a random variable of the count of tags mapped to  $r$ , and by  $x$  its particular realisation. If we denote by  $\lambda$ ,  $0 < \lambda < 1$ , the *proportion* (fraction) of reads in the library  $L$  with tags mappable to  $r$ , then  $X$  can be modelled as a binomial random variable,  $Bin(\lambda, |R|)$ :

$$\mathbb{P}[x = X(R)] = \binom{|R|}{x} \lambda^x (1 - \lambda)^{|R| - x}, \quad (1)$$

for  $x = 0, 1, \dots, |R|$ . In a typical case of interest  $\lambda \ll 1$  and the distribution of  $X$  is very well approximated by the Poisson distribution [10],  $Poi(\mu)$ :

$$\mathbb{P}[x = X] = \mu^x \frac{e^{-\mu}}{x!}, \quad (2)$$

where  $x = 0, 1, \dots, |R|$  and the *Poisson rate* is defined as  $\mu = \lambda|R|$ .

## 3 The Poisson Margin Test

Now we focus on comparing two libraries,  $L_A$  and  $L_B$ , from which random samples of mappable tags  $R_A$  and  $R_B$  were drawn, respectively. Suppose we

have observed counts  $x_A$  and  $x_B$  of tags mapped to the specific peak range  $r$  of interest, and  $\lambda_A$  and  $\lambda_B$  are the corresponding proportions of tags in the libraries  $L_A$  and  $L_B$  mappable to  $r$ , respectively. Let  $(a, b) \in \{(A, B), (B, A)\}$  and the following relation for empirical proportions holds:

$$\hat{\lambda}_a := \frac{x_a}{|R_a|} < \hat{\lambda}_b := \frac{x_b}{|R_b|}. \tag{3}$$

How strong is this evidence for  $\lambda_a < \lambda_b$ ? We approach this issue in a typical statistical hypothesis testing manner. Namely, we are interested in testing the *alternative hypothesis (H1)* that  $\lambda_a < \lambda_b$  versus the *null hypothesis (H0)* that  $\lambda_a \geq \lambda_b$ , i.e. that the complementary relation holds. As a natural test statistic we can use the maximal probability of observing at least as extreme counts  $X_a, X_b$  under the null hypothesis (H0). This probability we shall quantify in two different ways, a test based on the binomial distribution (*Binomial Margin,  $\mathcal{M}_{Bi}$* ) and its Poisson approximation (*Poisson Margin,  $\mathcal{M}_{Po}$* ), respectively:

$$\begin{aligned} \mathcal{M}_{Bi}(x_a, x_b) &:= \sup_{\lambda_a \geq \lambda_b > 0} \mathbb{P} [X_a \leq x_a \ \& \ x_b < X_b \mid X_i \sim Bin(\lambda_i, |R_i|)], \\ \mathcal{M}_{Po}(x_a, x_b) &:= \sup_{\lambda_a \geq \lambda_b > 0} \mathbb{P} [X_a \leq x_a \ \& \ x_b < X_b \mid X_i \sim Poi(\lambda_i |R_i|)], \end{aligned} \tag{4}$$

assuming the observed proportions relation (3) holds and, otherwise:

$$\mathcal{M}_{Bi}(x_a, x_b) = \mathcal{M}_{Po}(x_a, x_b) := 1. \tag{5}$$

In practice both tests are numerically equivalent, but  $\mathcal{M}_{Po}$  is easier to handle analytically and computationally, and will be the primary focus of the rest of this paper.

We observe that the above definitions do not require that the sample sizes  $|R_A|$  and  $|R_B|$  be equal. By definition, the Poisson margin  $\mathcal{M}_{Po}$  is the tightest universal upper bound on the following probability

$$\mathbb{P} [X_a \leq x_a \ \& \ x_b < X_b \ \& \ \lambda_a \geq \lambda_b \mid X_i \sim Poi(\lambda_i |R_i|)] \leq \mathcal{M}_{Po}(x_a, x_b).$$

In this sense it is a very conservative  $p$ -value, corresponding to the worst case scenario test.

### 3.1 Computation of Poisson Margin

The following result facilitates the efficient numerical evaluation of  $\mathcal{M}_{Po}$ ; the proof is presented in the on-line Supplement [1]. Let

$$\rho := |R_a|/|R_b|, \text{ and } \chi := x_a/x_b.$$

**Theorem 1.** *If the empirical relation for proportions (3) holds, then*

$$\mathcal{M}_{Po}(x_a, x_b) = \sup_{0 < \mu} e^{-2\mu} \sum_{i=0}^{x_a} \frac{(2\mu)^i}{i!(1+\rho)^i} \sum_{j>x_b} \frac{(2\mu\rho)^j}{j!(1+\rho)^j}, \tag{6}$$

where the supremum is achieved for  $\mu = \mu_*$ , the only solution of the reduced critical equation:

$$0 = E(\mu) := \rho + \rho \sum_{i=1}^{x_a} \prod_{j=0}^{i-1} \frac{(1 + \rho)(x_a - j)}{2\mu} - \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{2\mu\rho}{(1 + \rho)(x_b + j)} \quad (7)$$

with the function  $E$  monotonically decreasing for  $\mu > 0$ , from  $+\infty$  to  $-\infty$ . Moreover, if Eqn. 3 holds, then

$$\frac{\rho - \chi + \sqrt{(\rho - \chi)^2 + 4\chi\rho(1 + \rho)}}{4\rho} \leq \frac{\mu_*}{x_b} \leq \frac{(1 + \chi)(1 + \rho)}{2\rho + 1} + O\left(\frac{1}{x_b}\right), \quad (8)$$

where we use the “ $O$ ”-notation for the negligible rounding term such that

$$\lim_{\varepsilon \rightarrow 0} |O(\varepsilon)/\varepsilon| < \chi/4 + 1/2.$$

Equation (7) is easy to solve numerically, using Newton’s method for example, even for large counts where  $x_a, x_b \sim 10^5$ . The function  $E(\mu)$  is monotonically decreasing and the bounds (8) can be used for initialisation of the solver iterations. The sums in (7) have quickly decaying terms, so in practice they are reduced to a summation of only a few terms. One of the aims of our derivation was to develop such a simplification and to remove some very small nuisance factors that are below the computer precision, say with  $\log_{10}$  below  $-308$  (= the limit of IEEE-754 double precision).

*Proof Outline.* We express (4) explicitly as a 2-dimensional optimisation task:

$$\mathcal{M}_{Po}(x_a, x_b) = \sup_{0 < \lambda_b \leq \lambda_a} e^{-\mu_a - \mu_b} \sum_{i=0}^{x_a} \frac{\mu_a^i}{i!} \sum_{j > x_b} \frac{\mu_b^j}{j!} \Bigg|_{\substack{\mu_a = \lambda_a |R_a| \\ \mu_b = \lambda_b |R_b|}}, \quad (9)$$

which can be reduced to the 1-dimensional optimisation

$$\mathcal{M}_{Po}(x_a, x_b) = \sup_{0 < \lambda} e^{-\mu_a - \mu_b} \sum_{i=0}^{x_b} \frac{\mu_a^i}{i!} \sum_{j > x_b} \frac{\mu_b^j}{j!} \Bigg|_{\substack{\mu_a = \lambda |R_a| \\ \mu_b = \lambda |R_b|}}.$$

The latter task can be solved by finding a solution  $\lambda_*$  of the critical equation for the function of  $\lambda$  under “sup” above, which after some simplifications, introduction of variable  $\mu := \lambda \frac{|R_a| + |R_b|}{2}$  and removal of small positive factors reduces to (7). The details are available on-line [11]. □

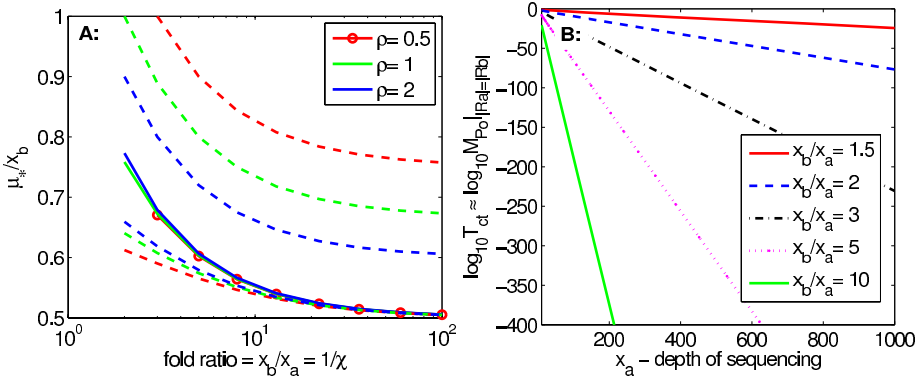
Figure 2.A shows the numerical evaluation of  $\mathcal{M}_{Po}$  across a range of counts which occur in practice. This figure clearly shows where the effects of the precision limits of IEEE-754 become apparent: the significant, “dark blue” shaded part of the plot corresponds to  $p$ -values  $< 10^{-400}$ . The thousands of peaks in the experimental data discussed in Section 4 fall into this region, see [11]. Note that the truncation of  $\log_{10} \mathcal{M}_{Po}$  at  $-500$  is used in Figure 2.A purely for the purpose of visualisation.

### 3.2 Related Statistical Tests

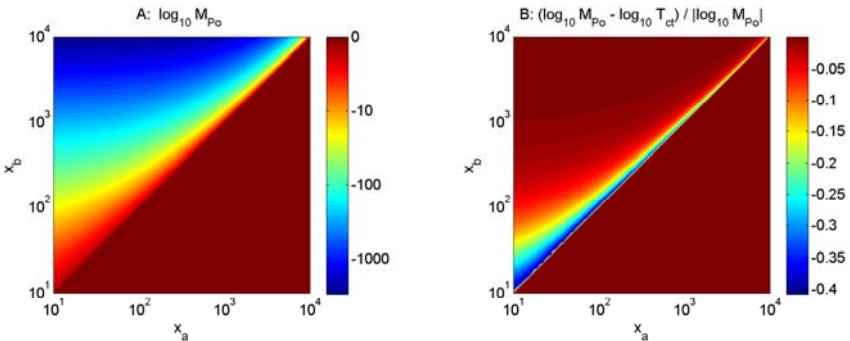
The Poisson Margin test is closely related to some other statistical tests. This relationship has been explored elsewhere (see [11, 5]) and here we mention only the Counts test and the Coin Toss test.

For  $x_a < x_b$  the Counts test is defined as

$$T_*(x_a, x_b) := \sup_{\mu_a \geq \mu_b > 0} \mathbb{P} [X_a \leq x_a \ \& \ x_b < X_b \mid X_i \sim Poi(\mu_i)] \quad (10)$$



**Fig. 1. A:** Lower and upper bounds on  $\mu_*$  given by Eqn. (8) of Theorem 1 (broken lines) and compared to the exact values  $\mu_*$  given by solution of Eqn. 7; here we show averages for  $\mu_*$  as solid lines, the values for evaluation over  $x_a$ -grid of 100 values logarithmically spaced between 1 and 1000 and corresponding  $x_b := x_a/\chi < 10,000$ . **B:** Computational validation of the Scaling Power Law given by Theorem 3. The plots show clearly the asymptotical power scaling law (12):  $T_{ct}(\kappa x_a, \kappa x_b) \approx T_{ct}(x_a, x_b)^\kappa$  translating to the linear dependence in the plots of the form  $x_a \mapsto \log_{10} T_{ct}(x_a, x_a/\chi) \approx x_a \times A^{-1} \log_{10} T_{ct}(A, A/\chi)$ , where  $A \gg 1$  is a constant.



**Fig. 2. A:** Numerical evaluation of  $\log_{10} \mathcal{M}_{Po}(x_a, x_b)$  for the case of  $|R_a| = |R_b|$  and **B:** of the relative difference  $(\log_{10} \mathcal{M}_{Po} - \log_{10} T_{ct}) / |\log_{10} \mathcal{M}_{Po}|$ . The evaluation has been done for the regular logarithmic grid of count values  $1 \leq x_a, x_b \leq 10,000$ .



for  $x_a < x_b$  and the Coin Toss test as

$$\mathcal{T}_{ct}(x_a, x_b) := \mathbb{P} \left[ X \leq x_a \mid X \sim \text{Bin} \left( \frac{1}{2}, x_a + x_b \right) \right]. \tag{11}$$

The Coin Toss test was used in [3][2]. The data from the latter paper will be used for our experimental validation, hence, indirectly, we will be comparing against this statistic, and a clarification of its relationship to the Poisson Margin is appropriate here. Both papers have used Coin Toss for computing the statistical significance of the difference between two counts  $x_a$  and  $x_b$  in the same fashion as in the case of  $\mathcal{M}_{Po}$ , but for the special case when libraries have equal, or equalised, sizes, namely  $|R_a| \approx |R_b|$ . In this special case they are “equivalent” to the  $\mathcal{M}_{Po}$  test in the following sense.

**Theorem 2.** *If  $|R_a| = |R_b|$  and the empirical proportion relation (3) holds, then*

$$\mathcal{M}_{Po}(x_a, x_b) = T_*(x_a, x_b) \approx \mathcal{T}_{ct}(x_a, x_b).$$

*Proof outline.* If  $|R_a| = |R_b|$ , then  $\lambda_a < \lambda_b$  is equivalent to  $\mu_a := \lambda_a |R_a| < \mu_b := \lambda_b |R_b|$ , hence the equivalence  $\mathcal{M}_{Po}(x_a, x_b) = T_*(x_a, x_b)$  follows from the definitions (4) and (10). The approximation by  $\mathcal{T}_{ct}$  has been argued in [5]; here we demonstrate it by numerical evaluation presented in Figure 2.  $\square$

Figure 2.B shows a numerical evaluation of the differences between  $\log_{10} \mathcal{M}_{Po}|_{|R_a|=|R_b|}$  and  $\log_{10} \mathcal{T}_{ct}$  over a grid of values of  $x_a$  and  $x_b$  which occur in practice. Note the differences in the shading scales used in the two sub-figures. The difference  $|\log_{10} \mathcal{M}_{Po} - \log_{10} \mathcal{T}_{ct}|$  is  $< 2$ , hence in the areas of significant values of  $\log_{10} \mathcal{M}_{Po}$ , say  $< -100$ , it composes practically negligible correction of  $\leq 1\%$ . This also allows the extension of the scaling properties  $\mathcal{T}_{ct}$  discussed below to the case of  $\mathcal{M}_{Po}|_{|R_a|=|R_b|}$ .

### 3.3 Scaling Properties

The following result can be shown formally (see [5]).

**Theorem 3 (Scaling Power Law).** *Let  $0 < 2x_a < x_b$  be two integers and  $\kappa > 1$ . Then*

$$\log \mathcal{T}_{ct}(\kappa x_a, \kappa x_b) = \kappa \log \mathcal{T}_{ct}(x_a, x_b) + \frac{o(x_b)}{x_b}, \tag{12}$$

where  $\frac{o(x_b)}{x_b} \rightarrow 0$  for  $x_b \rightarrow \infty$  denotes a ‘negligible’ correction.

The computational validation of this result and implied practical extension to the whole range of values  $x_b > 0$  is presented in Figure 11.B which is sufficient for our discussion below. Plots in there show clearly the asymptotical power scaling law (12):  $\mathcal{T}_{ct}(\kappa x_a, \kappa x_b) \approx \mathcal{T}_{ct}(x_a, x_b)^\kappa$  translating to the linear dependence in the plots of the form  $\log_{10} \mathcal{T}_{ct}(x_a, x_a/\chi) \approx x_a \times A^{-1} \log_{10} \mathcal{T}_{ct}(A, A/\chi)$ , where  $A \gg 1$  is a constant.

The above Theorem and Figure 1B facilitate a discussion of two important issues in the analysis of the NGS data, namely, (i) the impact of the number of lanes used by the sequencing machine to map the library and (ii) scaling of the counts, in the case of typically unequal sizes of the sequenced and then mapped tag sets. Firstly, they tell us that having  $\kappa$  times more reads sequenced, e.g. using  $\kappa$  lanes in the sequencing machine rather than one, will provide an exponentially stronger (i.e. smaller, exponentiated by  $\kappa$ )  $p$ -values, asymptotically for large  $x_b$ . This can be used, for example, as guidance for selection of more or fewer lanes in an NGS experiment.

However, those results also point to fragility of “count scaling”. More precisely, if the numbers of reads actually sequenced and mapped are significantly different,  $|R_a| \not\approx |R_b|$ , then either we need statistical tests that are intrinsically immune to such differences or we need to normalize the counts in a candidate peak region to make them comparable. The  $\mathcal{M}_{Po}$  test (4) falls in the first category while the  $\mathcal{T}_{ct}$  test represents the latter.

## 4 Experimental Validation

It is far from clear that the postulated statistical test will provide useful results in practice. Simply put, the worst case scenario embraced in definition (4) may be too conservative and the generated  $p$ -values too close to 1 to be informative. In order to address this concern, we have decided to focus on the well studied NGS application of ChIP-Seq peak calling, which involves comparison of only two libraries (the target and the dedicated control). The more complex problem of differential analysis of multiple libraries will be addressed in future work.

In order to validate our method we have used public domain ChIP-Seq data. In particular, [2] provides 36,998 putative locations/regions for binding STAT1 and 24,739 locations/regions for Pol II. Note that both databases in [2] have been used as the basis of the most recent ENCODE data in their corresponding domains, hence we have no alternative ‘gold standards’ to evaluate the results of our analysis. In this paper we deal with this obstacle by using a procedure evaluating the results of analysis by checking their internal consistency. This procedure is outlined below in Section 4.3.

Since [2] also provides the Eland mappings of the tags for the controls, STAT1 and Pol II data sets we have performed an additional independent analysis of this data using Poisson Margin outlined in the following two sections.

### 4.1 Re-ordering

We have used the list of peak ranges, peak locations and range boundaries exactly as in [2]. For each range we have extracted (raw) counts, following the protocol described in the original paper and then used the  $\mathcal{M}_{Po}$  statistic to allocate the  $p$ -value (see Supplementary Table 1 for STAT1 and Table 2 for Pol II). Although the order according to the  $\mathcal{M}_{Po}$  method is only slightly different from the original, the overlap is  $> 90\%$ , see Table 1, the differences in performance benchmarks especially for STAT1 are significant.

## 4.2 De-novo Significant Range Location

In our experiments described below we have implemented and run a fixed size sliding window method across the genome comparing the number of tags in each window for the control and target samples, for each sample for both DNA strands pooled together. Regions of significance are then defined by thresholding the  $p$ -values obtained from the Poisson Margin test. This resembles the approach in [11]. This effectively separates the tasks of finding regions of the genome where we believe a peak lies and determining the location of the antibody binding site itself, the former task being done efficiently on the whole genome scale and the latter being done intensively on just those regions identified by the genome wide scan as containing a peak. The genome was scanned sequentially with a window of 200 bp, shifted every 4 bases, which took about 17 minutes on a workstation with 2GHz Opteron CPUs and 32Gb of main memory.

We have identified 35,229 peak ranges for STAT1 using un-adjusted  $p$ -values  $<1E-4$  and 28,890 using un-adjusted  $p$ -values  $<1E-6$  for Pol II, with thresholds chosen to match the numbers of peaks in the original publication. The range was defined as a contiguous region of 200 BP blocks which passed the threshold.

Such a procedure can be followed by refinements of boundaries and more precise location of the range boundaries and more precise peak location. Example of such secondary adjustments can be found in [2][11][12], but we used none here.

## 4.3 Genome Annotation Test

We wish to quantify consistency of the putative peak ranges by quantifying ability to predict those locations on the whole genome by a predictor trained on a part of the genome. In our experiments peak ranges from chromosome 22 were used for training and those from the remaining chromosomes were used only for testing. For quantifying prediction accuracy we have adapted protocol 1B in [13][14][15]. In brief, the genome is divided into 500bp non-overlapping segments (total of 5,362,342 segments not containing ‘N’s), with each segment labeled as positive if it overlaps a peak range and negative otherwise (the positive segments comprise  $< 1\%$  of the total number of segments). These labels were used to build a predictor and verify its performance measured by precision and recall. We recall that “Precision” means ratio of true positive retrieved (for a given decision threshold) to number of retrieved cases, and “Recall” is the number of true positive retrieved divided by the total number of true positive cases.

A linear support vector machine (*SVM*) was developed to label each 500bp segment independently [15]. Each 500bp segment was represented as a feature vector containing frequency counts of 4-mers contained within. Recursive feature elimination was used to reduce the model’s number of features and other meta-parameters were set to maximise the area under precision-recall curve (PRC) in an internal cross-validation on the training data.

This method works very well for some tasks such as the prediction of transcription start sites and the binding of some transcription factors (e.g., c-Myc) [15], and seems to significantly outperform other methods such as standard position

weight matrices (*PWM*). From our experience, STAT1 is one of the harder transcription factors to predict, however we still observe much higher performance using the SVM predictor than with PWMs [15].

### 4.4 Results

Table 1 and Figure 3 summarise results for few its variations of genome annotation experiment described above, for STAT1 and Pol II data, respectively. We recall, the data from chromosome 22 was used for training exclusively and test results reported are for data from the whole genome. The following four basic variations of experiment consisted in usage of different sets of peaks:

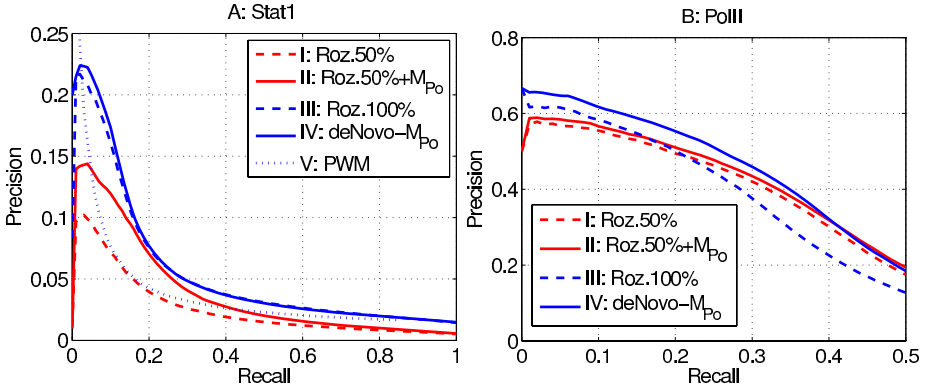
- I: The top 50% of the original list in [2];
- II: The top 50% of the list in [2] after sorting by  $\mathcal{M}_{Po}$  method;
- III: The whole 100% list in [2];
- IV: A *de novo* list of peaks derived as outlined in Section 4.2

Additionally, for STAT1 we have also tested a typical position weight (PWM) matrix from TRANSFAC ® 7.0. This is reported as row “V” in Table 1. In this case the “score” per range in [2] is defined as the max of PWM scores for all positions within a 500BP tile. The overlap in Table 1 was calculated using the top 37k tiles as scored by the PWM.

We observe that for variants I and II for STAT1, the area under the PRC curve (4.0%) is approximately 1.5 times that for the original [2] ordering (2.6%). For Pol II the corresponding difference is smaller, due to larger similarity of the ordered lists and much higher accuracy of predictions, but differences between *de-novo* (blue solid) and 100% list in [2] (broken blue) in Figure 3.B is well pronounced.

**Table 1.** Summary of ordered peaks lists overlap with the list in [2] and the accuracy prediction of binding site on the whole genome. In experiments I-IV we use data on chromosome 22 for training SVM exclusively. All values listed are in %.

Experiment	List overlap (%) with [2]				Prediction for whole genome		
	Number of top peaks				Area under Prec.-Rec. curve	Prec. at recall	
	10%	20%	50%	100%		10%	20%
<b>STAT1</b>							
<b>I:</b> Roz.50%	100	100	100	-	2.6	7.6	4.2
<b>II:</b> Roz.+ $\mathcal{M}_{Po}$ 50%	96	94	91	-	<b>4.0</b>	<b>12.3</b>	<b>7.5</b>
<b>III:</b> Roz.100%	100	100	100	100	5.5	17.1	8.0
<b>IV:</b> deNovo- $\mathcal{M}_{Po}$	68	69	70	83	<b>5.7</b>	<b>18.2</b>	<b>8.2</b>
<b>V:</b> Roz.100% PWM	3	5	9	12	4.1	8.4	4.4
<b>Pol II</b>							
<b>I:</b> Roz.50%	100	100	100	-	24.4	56.0	50.1
<b>II:</b> Roz.+ $\mathcal{M}_{Po}$ 50%	53	85	95	-	<b>25.6</b>	<b>57.4</b>	<b>51.7</b>
<b>III:</b> Roz.100%	100	100	100	100	23.2	58.8	51.2
<b>IV:</b> deNovo- $\mathcal{M}_{Po}$	66	71	74	77	<b>26.9</b>	<b>62.4</b>	<b>56.1</b>



**Fig. 3.** Precision–Recall curves for STAT1 (A) and Pol II (B) corresponding to Table 1. We report test results for the whole genome for the variants I–V of the genome annotation test experiment described in Section 4.4.

## 5 Discussion

The basic protocol described in this paper can be extended in a number of directions. In the experiments we have used only uniquely mapped reads. The density of such reads varies along the genome, which can affect the relative  $p$ -values for peaks at different locations since both  $\mathcal{M}_{Po}$  and the Coin Tossing statistic ( $\mathcal{T}_{ct}$ ) [2] are sensitive to the sequencing depth (see Figure 1.B). A simple way around this obstacle is to scale up the observed counts (per range) inversely to the fraction of mappable tags for the region. Such information for uniquely mapped tags of length 30 is provided in [2], but information of uniquely mapped up to 2 mismatches (which we prefer) is not currently available (see comment in [2, Supplement]). We have not used this correction here.

We present an alternative to the method introduced previously, e.g. [2], [4] or [11]. We have focussed on the first of those references since it is one of the most recent and allows access to good quality of experimental data, included in ENCODE. We have shown that a principled analysis of such data using our method is feasible with minimal need for (arbitrary) design choices and with a minimal number of data-adjustable parameters. (An illustrating example here is the introduction in [2, p. 73] of an ad hoc parameter  $0 \leq P_f \leq 1$  for the fraction of putative highest peaks to be excluded from regression for local normalisation of counts). Our approach is robust, and can be applied to a wide variety of experimental designs involving different numbers of samples, possibly from different cell lineages. It is applicable precisely because it does not require scaling of the individual libraries of reads. The results in [5] and Section 3.3 show that such a scaling, if applied, should be done with extreme caution, if the analysis is to be meaningful.

## 6 Conclusions

We have developed a principled statistical test for the detection of significant reads concentrations which is directly applicable to libraries of different (unmatched) sizes without any scaling of read counts and have demonstrated that such a scaling could introduce significant bias in the computed  $p$ -values. Although our statistical test targets differential analysis for multiple NGS libraries, the initial validation in this paper is restricted to the simplest case of comparison of a target library to a matching reference. Using the recent Encode Chip-Seq data we have shown that our test delivers non-vacuous results, with peak calling accuracies comparable or even improved with respect to the the original dedicated algorithm. The absence of adequate gold standards for benchmarking was circumvented by application of a novel internal consistency check based on the accuracy of generalisation of a supervised learning predictor. Demonstration of the utility of that protocol is the second major contribution of this paper.

## Acknowledgements

NICTA is funded by the Australian Government's Department of Communications, Information Technology and the Arts, the Australian Research Council through Backing Australia's Ability, and the ICT Centre of Excellence programs.

## References

1. Kowalczyk, A., Bedo, J., Conway, T., Beresford-Smith, B.: Poisson Margin Test for Normalisation Free Significance Analysis of NGS Data - Supplementary Materials (2009), <http://www.genomics.csse.unimelb.edu.au/peakfiltsup>
2. Rozowsky, J., Euskirchen, G., Auerbach, R., Zhang, Z., Gibson, T., Bjornson, R., Carriero, N., Snyder, M., Gerstein, M.: Peakseq enables systematic scoring of chip-seq experiments relative to controls. *Nature Biotechnology* 27, 66–75 (2009)
3. Nix, D., Courdy, S., Boucher, K.: Empirical methods for controlling false positives and estimating confidence in chip-seq peaks. *BMC Bioinformatics* 9, 523 (2008)
4. Robertson, G., Hirst, M., Bainbridge, M., Bilenky, M., Zhao, Y., Zeng, T., Euskirchen, G., Bernier, B., Varhol, R., Delaney, A., et al.: Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods* 4, 651–657 (2007)
5. Kowalczyk, A.: Some Formal Results for Significance of Short Read Concentrations (2009), <http://www.genomics.csse.unimelb.edu.au/shortreadtheory>
6. Baggerly, K.A., Deng, L., Morris, J.S., Aldaz, C.M.: Differential expression in SAGE: accounting for normal between-library variation. *Bioinformatics* 19, 1477–1483 (2003)
7. Robinson, M., Smyth, G.: Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics* 23(21), 2881–2887 (2007)
8. Bloushtain-Qimron, N., Yao, J., Snyder, E.: Cell type-specific dna methylation patterns in the human breast. *PANS* 105, 14076–14081 (2008)
9. Zang, C., Schones, D.E., Zeng, C., Cui, K., Zhao, K., Peng, W.: A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics* 25, 1952–1958 (2009)

10. Keeping, E.: Introduction to Statistical Inference. Dover, New York (1995) ISBN 0-486-68502-0; Reprint of 1962 edition by D. Van Nostrand Co., Princeton, New Jersey
11. Zhang, Y., Liu, T., Meyer, C., Eeckhoutte, J., Johnson, D., Bernstein, B., Nussbaum, C., Myers, R., Brown, M., Li, W., Liu, X.S.: Model-based analysis of chip-seq (macs). *Genome Biology* 9(9), R137 (2008)
12. Ji, H., Jiang, H., Ma, W., Johnson, D., Myers, R., Wong, W.: An integrated software system for analyzing ChIP-chip and ChIP-seq data. *Nature Biotechnology* 26, 1293–1300 (2008)
13. Sonnenburg, S., Zien, A., Ratsch, G.: Arts: accurate recognition of transcription starts in human. *Bioinformatics* 22, e423–e480 (2006)
14. Abeel, T., Van de Peer, Y., Saeys, Y.: Toward a gold standard for promoter prediction evaluation. *Bioinformatics* 25, i313–i320 (2009)
15. Bedo, J., MacIntyre, G., Haviv, I., Kowalczyk, A.: Simple SVM based whole-genome Segmentation (2009), Available from Nature Precedings <http://dx.doi.org/10.1038/npre.2009.3811.1>

# Compressing Genomic Sequence Fragments Using SLIMGENE

Christos Kozanitis<sup>1</sup>, Chris Saunders<sup>2</sup>, Semyon Kruglyak<sup>2</sup>,  
Vineet Bafna<sup>1</sup>, and George Varghese<sup>1</sup>

<sup>1</sup> University of California San Diego, La Jolla CA 92093, USA

<sup>2</sup> Illumina Inc, San Diego CA 92121, USA

**Abstract.** With the advent of next generation sequencing technologies, the cost of sequencing whole genomes is poised to go below \$1000 per human individual in a few years. As more and more genomes are sequenced, analysis methods are undergoing rapid development, making it tempting to store sequencing data for long periods of time so that the data can be re-analyzed with the latest techniques. The challenging open research problems, huge influx of data, and rapidly improving analysis techniques have created the need to store and transfer very large volumes of data.

Compression can be achieved at many levels, including trace level (compressing image data), sequence level (compressing a genomic sequence), and fragment-level (compressing a set of short, redundant fragment reads, along with quality-values on the base-calls). We focus on fragment-level compression, which is the pressing need today.

Our paper makes two contributions, implemented in a tool, SLIMGENE. First, we introduce a set of domain specific loss-less compression schemes that achieve over 40× compression of fragments, outperforming bzip2 by over 6×. Including quality values, we show a 5× compression using less running time than bzip2. Second, given the discrepancy between the compression factor obtained with and without quality values, we initiate the study of using ‘lossy’ quality values. Specifically, we show that a lossy quality value quantization results in 14× compression but has minimal impact on downstream applications like SNP calling that use the quality values. Discrepancies between SNP calls made between the lossy and lossless versions of the data are limited to low coverage areas where even the SNP calls made by the lossless version are marginal.

## 1 Introduction

With the advent of next generation sequencing technologies [8,15,16,10], the cost of sequencing whole genomes has decreased dramatically in the past several years, and is poised to go below \$1000 per human individual in a few years. As more and more genomes are sequenced, researchers are faced with the daunting challenge of interpreting all of the data. At the same time, analysis methods are undergoing rapid development making it tempting to store sequencing data for long periods of time so that the data can be re-analyzed with the latest techniques. The challenging open research problems, huge influx of data, and rapidly improving analysis techniques have created the need to store and transfer very large volumes of data.



The study of human variation, and genome wide association (GWAS) was traditionally accomplished using micro-arrays, for which the data is smaller by over 3 orders of magnitude. However, these GWA studies have been able to explain only a very small fraction of heritable variation present in complex diseases. Many researchers believe that whole genome sequencing may overcome some of the limitation of micro-arrays. The incomplete picture formed by micro-arrays, the many applications of sequencing (Ex: structural variations), and the expected improvement in cost and throughput of sequencing technology ensure that sequencing studies will continue to expand rapidly. The question of data handling must therefore be addressed.

Even with the limited amount of genetic information available today, sites such as the Broad Institute and the European Bioinformatics institute are among the biggest storage consumers in the world, spending millions of dollars on storage [6]. Beyond research laboratories, the fastest growing market for sequencing studies is big pharmaceutical companies [6]. Further, population studies on hundreds of thousands of individuals in the future will be extremely slow if individual disks have to be shipped to an analysis center. The *single* genome data set we use for our experiments takes 285GB in uncompressed form. At a network download rate of 10Mb/s this data set would take 63.3 hours to transfer over the Internet. In summary, reducing storage costs and improving interactivity for genomic analysis makes it imperative to look for ways to compress genomic data.

While agnostic compression schemes like Lempel-Ziv [20] can certainly be used, we ask if we can exploit the specific domain to achieve better compression. As an example, domain-specific compression schemes like MPEG-2 [13] exploit the use of a dictionary or reference specific to the domain. Here, we exploit the fact that the existing human assembly can be used as a reference for encoding. We mostly consider loss-less compression algorithms. Specifically, given a set of genomic data  $S$ , we define a compression algorithm by a pair of functions  $(\mathcal{C}, \mathcal{D})$  such that  $\mathcal{D}(\mathcal{C}(S)) = S$ . The compression factor *c.f.*, defined by  $|S|/|\mathcal{C}(S)|$  describes the amount of compression achieved.

The genomic data  $S$  itself can have multiple forms and depends upon the technology used. Therefore, the notion of ‘loss-less’ must be clarified in context. In the Illumina Genome Analyzer, each cycle produces 4 images, one for each of the nucleotides; consequently, the set  $S$  consists of the set of all images in all cycles. By contrast, the ABI technology maps adjacent pairs of nucleotides to a ‘color-space’ in the unprocessed stage. We refer to compression at this raw level as *a. Trace Compression*. The raw, trace data is processed into base-calls creating a set of fragments (or, reads). This processing may have errors, and a quality value (typically a Phred-like score given by  $-\lfloor 10 \log(P_{\text{Error}}) \rfloor$ ) is used to encode the confidence in the base-call. In *b. Fragment Compression*, we define the genomic data  $S$  by the set of reads, along with quality values of each base-call. Note that the set of reads all come from the genomic sequence of an individual. In *c. Sequence Level Compression*, we define the set  $S$  simply as the diploid genome of the individual.

There has been some recent work on compressing at the sequence level [24, 5, 12]. Brandon and colleagues introduce the important notion of maintaining differences against a genomic reference, and integer codes for storing offsets [2]. However, such

sequence compression relies on having the fragments reconciled into a single (or diploid) sequence. While populations of entire genomes are available for mitochondria, and other microbial strains sequenced using Sanger reads, current technologies provide the data as small genomic fragments. The analysis of this data is evolving, and researchers demand access to the fragments and use proprietary methods to identify variation, not only small mutations, but also large structural variations [9,7,18,14]. Further, there are several applications (e.g., identifying SNPs, structural variation) of fragment data that do not require the intermediate step of constructing a complete sequence.

Clearly, trace data are the lowest level data, and the most difficult to compress. However, it is typically accessed only by a few expert researchers (if at all), focusing on a smaller subset of fragments. Once the base-calls are made (along with quality values), the trace data is usually discarded.

For these reasons, we focus here on fragment level compression. Note that we share the common idea of compressing with respect to a reference sequence. However, our input data are a collection of potentially overlapping fragments (each, say 100 bps long) annotated with quality values. These lead to different compression needs and algorithms from [25] because fragment compression must address the additional redundancy caused by high coverage and quality values. Further, the compression must efficiently encode differences due to normal variation *and* sequencing errors, for the downstream researcher.

**Contribution:** Our paper makes two contributions, implemented in a tool, SLIMGENE. First, we introduce a set of domain specific loss-less compression schemes that achieve over  $40\times$  compression of fragments, outperforming bzip2 by over  $6\times$ . Including quality values, we show a  $5\times$  compression. Unoptimized versions of SLIMGENE run at comparable speeds to bzip2. Second, given the discrepancy between the compression factor obtained with and without quality values, we initiate the study of using ‘lossy’ quality values and investigate its effect on downstream applications. Specifically, we show that using a lossy quality value quantization results in  $14\times$  compression but has minimal impact on SNP calls using the CASAVA software. Less than 1% of the calls are discrepant, and we show that the discrepant SNPs are so close to the threshold of detection, that no miscalls can be attributed to lossy compression. While there are dozens of downstream applications and much work needs to be done to ensure that coarsely quantized quality values will be acceptable for users, our paper suggests this is a promising direction for investigation.

## 2 Data-Sets and Generic Compression Techniques

**Generic compression techniques:** Consider the data as a string over an alphabet  $\Sigma$ . We consider some generic techniques. First, we use a reference string so that we only need to encode the differences from the reference. As each fragment is very small, it is critical to encode the differences carefully. Second, suppose that the letters of  $\sigma \in \Sigma$  are distributed according to probability  $P(\sigma)$ . Then, known compression schemes (Ex: Huffman codes, Arithmetic codes) encode each symbol  $\sigma$  using  $\log_2 \frac{1}{p(\sigma)}$  bits, giving an average of  $\mathcal{H}(P)$  (entropy of the distribution) bits per symbol, which is optimal for the distribution, and degrades to  $\log(|\Sigma|)$  bits in the worst case.

Our goal is to devise an encoding (based on domain specific knowledge) that minimizes the entropy. In the following, we will often use this scheme, describing the suitability of the encoding by providing the entropy values. Also, while it is asymptotically perfect, the exact reduction is achievable only if the probabilities are powers of 2. Therefore, we often resort to techniques that mimic the effect of Huffman codes. Finally, if there are inherent redundancies in data, we can compress by maintaining pointers to the first occurrence of a repeated string. This is efficiently done by tools such as bzip2, and we reuse the tools.

**Data formats:** Many formats have been proposed for packaging and exporting genomic fragments, including the SAM/BAM format [11,17], and the Illumina Export format [3]. Here, we work with the Illumina Export format, which provides a standard representation of Illumina data. It is a tab delimited format, in which every row corresponds to a read, and different columns provide qualifiers for the read. These include ReadID, CloneID, fragment sequence, and a collection of quality values. In addition, the format also encodes information obtained from aligning the read to a reference, including the chromosome strand, and position of the match. The key thing to note is that the fragment sequences, the quality values, and the match to the chromosomes represent about 80% of the data, and we will focus on compressing these. In SLIMGENE, each column is compressed independently, and the resulting data is concatenated.

### Data Sets: Experimental, and Simulated

We consider a data-set of human fragments, obtained using the Illumina Genome Analyzer, and mapped to the reference (NCBI 36.1, Mar. 2006). A total of  $1.1B$  reads of length 100 were mapped, representing  $35\times$  base-coverage of the haploid genome. We refer to this data-set as GAHUM. The fragments differ from the reference either due to sequencing errors or genetic variation, but we refer to all changes as errors. The number of errors per fragment is distributed roughly exponentially, with a heavy tail, and a mean of 2.3 errors per fragment, as shown below. Because of the heavy tail, we did not attempt to fit the experimental data to a standard distribution.

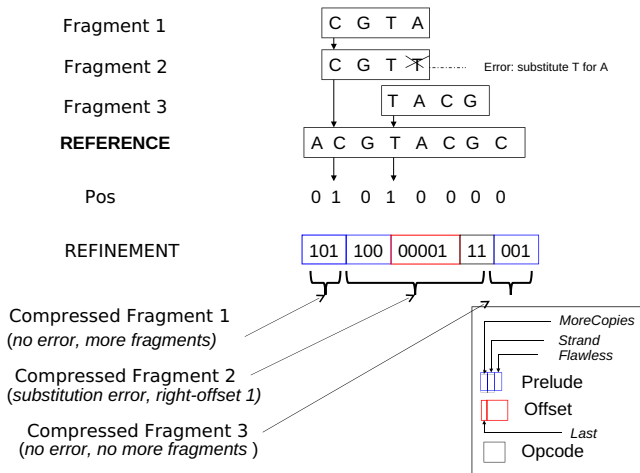
#Errors(k)	0	1	2	3	4	5	6	7	8	9	$\geq 10$
Pr( $k$ errors)	0.43	0.2	0.09	0.06	0.04	0.03	0.02	0.02	0.01	0.01	0.09

**Simulating coverage:** While we show all compression results on GAHUM, the results could vary on other data-sets depending upon the quality of the reads, and the coverage. To examine this dependence, we simulated data-sets with different error-rates, and coverage values. We choose fragments of length 100 at random locations from Chr 20, with a read coverage given by parameter  $c$ . To simulate errors, we use a single parameter  $P_0$  as the probability of 0 errors in the fragment. For all  $k > 0$ , the probability of a fragment having exactly  $k$  errors is given by  $P_k = \lambda \text{Pr}(k \text{ errors})$  from the distribution above. The parameter  $\lambda$  is adjusted to balance the distribution ( $\sum_i P_i = 1$ ). The simulated data-set with parameters  $c, P_0$  is denoted as GASIM( $c, P_0$ ).

### 3 Compressing Fragment Sequences

Consider an experiment with sequence coverage  $c$  ( $\sim 30\times$ ), which describes the expected number of times each nucleotide is sequenced. Each fragment is a string of characters of length  $L$  ( $\simeq 100$ ) over the nucleotide alphabet  $\Sigma$  ( $\Sigma = \{A, C, G, T, N\}$ ). The naive requirement is  $8c$  bits per nucleotide, which could be reduced to  $c \log(|\Sigma|) \simeq 2.32c$  bits with a more efficient encoding. We describe an encoding based on comparison to a reference that all fragments have been mapped to.

**The position vector:** Assume a *Position* bit vector POS with one position for every possible location of the human genome. We set  $POS[i] = 1$  if at least one fragment maps to position  $i$  ( $POS[i] = 0$  otherwise). For illustration, imagine an 8-character reference sequence, ACGTACGC, as depicted in Figure 1. We consider two 4bp fragments, CGTA and TACG, aligned to positions 2 and 4, respectively, with no error. Then,  $POS = [0, 1, 0, 1, 0, 0, 0, 0]$ . The bit vector POS would suffice if (a) each fragment matched perfectly (no errors), (b) matches to the forward strand and (c) at most one fragment aligns to a single position (possible if  $L > c$ ). The space needed reduces to 1 bit per nucleotide, (possibly smaller with a compression of POS), independent of coverage  $c$ .



**Fig. 1.** A simple proposal for fragment compression starts by mapping fragments to a reference sequence. The fragments are encoded by a Position Vector and a Refinement Vector consisting of variable size records representing each compressed fragment. The compressed fragments are encoded on a “pay as needed” basis in which more bits are used to encode fragments that map with more errors.

In reality, these assumptions are not true. For example, two Fragments 1 and 2 match at position 2, and Fragment 2 matches with a substitution (Figure 1). We use a *Refinement* vector that adds count and error information. However, the Refinement vector is designed on a “pay as needed basis” – in other words, fragments that align with fewer errors and fewer repeats need fewer bits to encode.

**The refinement vector:** The Refinement Vector is a vector of records, one for each fragment, each entry of which consists of a static *Prelude* (3 bits) and an *ErrorInstruction* record, with a variable number of bits for each error in the alignment.

The 3-bit *Prelude* consists of a *MoreCopies*, a *Strand* and a *Flawless* bit. All fragments that align with the same location of the reference genome are placed consecutively in the Refinement Vector and their *MoreCopies* bits share the same value, while the respective bits of consecutive fragments that align to different locations differ. Thus, in a set of fragments, the *MoreCopies* bit changes value when the chromosome location varies. The *Strand* bit is set to 0 if the fragment aligns with the forward strand and 1 otherwise, while the *Flawless* bit indicates whether the fragment aligns perfectly with the reference, in which case there is no following *ErrorInstruction*.

When indicated by the *Flawless* bit, the *Prelude* is followed by an *ErrorInstruction*, one for every error in the alignment. The *ErrorInstruction* consists of an *Offset* code (# bp from the last erroneous location), followed by a variable length *Operation Code* or *OpCode* field describing the type of error.

**Opcode:** As sequencing errors are mostly nucleotide substitutions, the latter are encoded by using 2 bits, while the overhead of allocating more space to other types of error is negligible. Opcode 00 is reserved for other errors. To describe all substitutions using only 3 possibilities, we use the circular chain  $A \rightarrow C \rightarrow G \rightarrow T \rightarrow A$ . The opcode specifies the distance in chain between the original and substituted nucleotide. For example, an  $A$  to  $C$  substitution is encoded as 01. Insertions, deletions, and counts of  $N$  are encoded using a Huffman-like code, to get an average of  $T = 3$  bits for Opcode.

**Offset:** Clearly, no more than  $O = \log_2 L$  bits are needed to encode the offset. To improve upon the  $\log_2(100) \simeq 7$  bits per error, note that the quality of base calling worsens in the later cycles of a run. Therefore, we use a *back-to-front* error ordering to exploit this fact, and a Huffman-like code to decrease  $O$ .

The record for Fragment 2 (CGTT, Figure III) provides an example for the error encoding, with a prelude equal to 100 (last fragment that maps to this location and error instructions follow) followed by a single *ErrorInstruction*. The next 5 bits (00001) indicate the relative offset of the error from the *end* of the fragment. The first bit of the offset is a “Last” bit that indicates that there are no more errors. The offset field is followed by an opcode (11) which describes a substitution of  $T$  for  $A$ , a circular shift of 3. Further improvement is possible.

**Compact Offset encoding:** Let  $\mathcal{E}$  denote the expected number of errors per fragment, implying an offset of  $\frac{L}{\mathcal{E}}$  bp. Instead, we use a single bitmap, ERROR, to mark the positions of all errors from all fragments. Second, we specify the error location for a given fragment as the number of bits we need to skip in ERROR from the start offset of the fragment to reach the error. We expect to see a ‘1’ after  $\max\{1, \frac{L}{c\mathcal{E}}\}$  bits in ERROR. Thus, instead of encoding the error offset as  $\frac{L}{\mathcal{E}}$  bp, we encode it as the count using

$$O = \log_2 \frac{L/\mathcal{E}}{\max\{1, \frac{L}{c\mathcal{E}}\}} = \min\{\log_2 \frac{L}{\mathcal{E}}, \log_2 c\}$$

bits. For smaller coverage  $c < \frac{L}{\mathcal{E}}$ , we can gain a few bits in computing  $O$ . Overall, the back-to-front ordering, and compact offset encoding leads to  $O \simeq 4$  bits.

**Compression analysis:** Here, we do a back-of-the-envelope calculation of compression gains, mainly to understand bottlenecks. Compression results on real data, and simulations will be shown in Section 3.1. To encode *Refinement*, each fragment contributes a *Prelude* (3 bits), followed by a collection of *Opcodes* ( $T$  bits each), and *Offsets* ( $O$  bits each). Let  $\mathcal{E}$  denote the expected number of errors per fragment, implying a refinement vector length of

$$\mathcal{E} \cdot (3 + T + O)$$

per fragment. Also, encoding POS, and ERROR requires 1 bit each, per nucleotide of the reference. The total number of bits needed per nucleotide of the reference is given by

$$2 + \frac{c}{L} \cdot \mathcal{E} \cdot (3 + O + T) \quad (1)$$

Substituting  $T = 3, O = 4, L = 100$ , we have

$$\text{c.f.} = \frac{8c}{2 + 0.1c\mathcal{E}} \quad (2)$$

Equation 2 provides a basic calculation of the impact of error-rate and coverage on compressibility using SLIMGENE. For GAHUM,  $\mathcal{E} = 2.3$  (Section 2). For high coverages, the c.f. is  $\simeq 8/0.23 \simeq 35$ . For lower coverages, the fixed costs are more important, but the POS and ERROR bitmaps are very sparse and can be compressed well, by (say) bzip2.

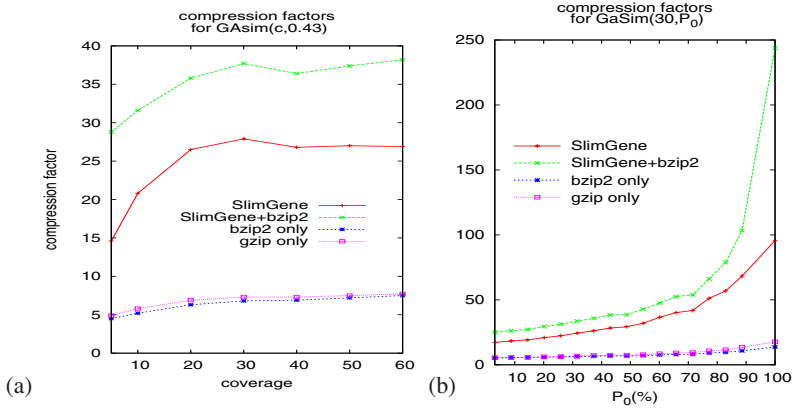
**Effectiveness:** The reader may wonder how our seemingly ad hoc encoding technique compares to information theoretic bounds. We first did an experiment to evaluate the effectiveness of OpCode assignment. We tabulated the probability of each type of error (all possible substitutions, deletions, and insertions) on our data set and used these probabilities to compute the expected OpCode length using our encoding scheme. We found that the expected OpCode length using our encoding was 2.97 which compares favorably with the entropy which was 1.9.

We also did an experiment to determine the effectiveness of the offset encoding. The width of the error location  $O$  depends on the number of bits that we need to skip in ERROR to reach the error location for a given fragment. We computed the error distribution of chromosome 20 of GAHUM and found that the majority of cases involved the skipping of no more than 10 bits in ERROR. Indeed, the entropy of the distribution of error offsets was 3.69. Thus, an initial allocation of 3 or 4 bits (with additional allocation as necessary) is reasonable.

### 3.1 Experimental Results on GASIM( $c, P_0$ )

We tested SLIMGENE on GASIM( $c, P_0$ ) to investigate the effect of coverage and errors on compressibility. Recall that for GAHUM,  $P_0 = 0.43, c = 30, \mathcal{E} = 2.3$ . As  $P_0$  is varied,  $\mathcal{E}$  is approximately  $\simeq \frac{2.3}{1-0.43} \cdot (1 - P_0) \simeq 4(1 - P_0)$ .

In Figure 2a, we fix  $P_0 = 0.43$ , and test compressibility of GASIM( $c, 0.43$ ). As suggested by Eq. 2, the compressibility of SLIMGENE stabilizes once the coverage is sufficient. Also, using SLIMGENE+bzip2, the compressibility for lower coverage is



**Fig. 2. Compressibility of GASIM( $c, P_0$ ).** (a) The compression factors achieved with change in coverage. c.f. is lower for lower coverage due to the fixed costs of POS, and ERROR, and stabilizes subsequently. (b) Compressibility as a function of errors. With high values of  $P_0$  (low error), up to 2 orders of magnitude compression is possible. Note that the values of  $P_0$  are multiplied by 100.

enhanced due to better compression of POS and ERROR. Figure 2b explores the dependency on the error rates using GASIM(30,  $P_0$ ). Again, the experimental results follow the calculations in Eq. 2, which can be rewritten as

$$\frac{8 \cdot 30}{2 + 0.1 \cdot 30 \cdot 4(1 - P_0)} \simeq \frac{20}{1 - P_0}$$

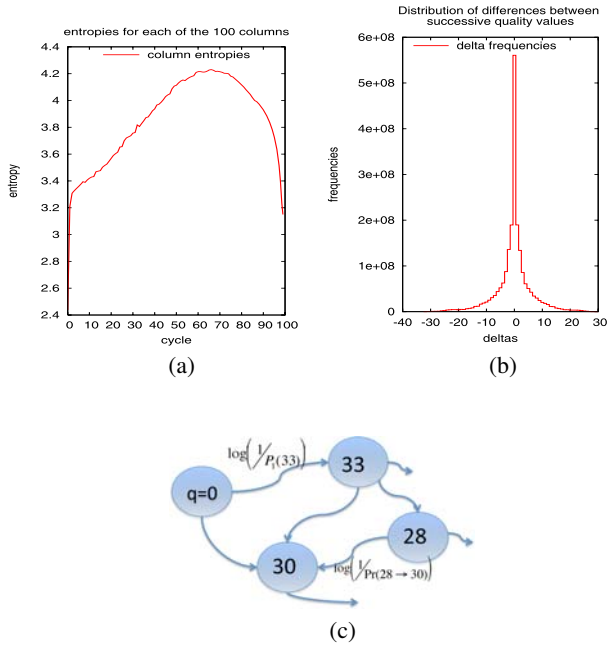
At high values of  $P_0$ , SLIMGENE produces 2 orders of magnitude compression. However, it outperforms bzip2 and gzip even for lower values of  $P_0$ .

## 4 Compressing Quality Values

For the Genome analyzer, as well as other technologies, the Quality values are often described as  $\approx -\log(P_{\text{ERR}})$ . Specifically, the Phred score is given to be  $\lfloor -10 \cdot \log(P_{\text{ERR}}) \rfloor$ . The default encoding for GAHUM require 8 bits to encode each  $Q$ -value. We started by testing empirically if there was a non-uniform distribution on these values (see Section 2). The entropy of the  $Q$ -values is 4.01. A bzip2 compression of the data-set resulted in 3.8 bits per  $Q$ -value. For further compression, we need to use some characteristics of common sequencers.

**Position dependent quality:** Base calling is highly accurate in the early cycles, while it gradually loses its accuracy in subsequent cycles. Thus, earlier cycles are populated by higher quality values and later cycles by lower values. To exploit this, consider a matrix in which each row corresponds to the  $Q$ -values of a single read in order. Each column therefore corresponds (approximately) to the  $Q$ -values of all reads in a single cycle. In Figure 3a, we plot the entropy of  $Q$ -value distribution at each columns. Not surprisingly, the entropy is low at the beginning (all values are high), and at the end (all values are low), but increases in the middle, with an average entropy of 3.85.





**Fig. 3. Distribution of quality, and  $\Delta$  values, and Markov encoding.** (a) A distribution of  $Q$ -values at each position. The entropy is low at the beginning (all values are high), and at the end (all values are low), but increases in the middle. (b) A histogram of  $\Delta$ -values. (c) Markov encoding: Each string of  $Q$ -values is described by a unique path in the automaton starting from  $q = 0$ , and is encoded by concatenating the code for each transition. Huffman encoding bounds the number of required bits by the entropy of the distribution. Edge labels describe the required number of bits for each transition.

**Encoding  $\Delta$  values:** The gradual degradation of the  $Q$ -values leads to the observation that  $Q$ -values that belong to neighboring positions differ slightly. Thus, if instead of encoding the quality values, one encodes their differences between adjacent values ( $\Delta$ ), it is expected that such a representation would be populated by smaller differences. For instance, Figure 3b shows a histogram of the distribution of  $\Delta$ -values. However, the entropy of the distribution is 4.26 bits per  $\Delta$ -value.

**Markov encoding:** We can combine the two ideas above by noting that the  $\Delta$ -values also have a Markovian property. As a simple example, assume that all  $Q$ -values from 2 to 41 are equally abundant in the empirical data. Then, a straightforward encoding would need  $\lceil \log_2(41 - 2 + 1) \rceil = 6$  bits. However, suppose when we are at quality value (say) 34 (Figure 3c), the next quality value is always one of 33, 32, 31, 30. Therefore, instead of encoding  $Q'$  using 6 bits, we can encode it using 2 bits, conditioning on the previous  $Q$ -value of 34.

We formalize this using a Markov model. Consider an automaton  $M$  in which there is a distinct node  $q$  for each quality value, and an additional start state  $q = 0$ . To start with, there is a transition from 0 to  $q$  with probability  $P_1(q)$ . In each subsequent step,



$M$  transitions from  $q$  to  $q'$  with probability  $\Pr(q \rightarrow q')$ . Using an empirical data-set  $D$  of quality values, we can estimate the transition probabilities as

$$\Pr(q \rightarrow q') = \begin{cases} 0 & (* \text{ if } q' = 0 *) \\ \text{fraction of reads with initial quality } q' & (* \text{ if } q = 0 *) \\ \frac{\#\text{pairs } (q, q') \text{ in } D}{\#\text{occurrences of } q \text{ in } D} & (* \text{ otherwise } *) \end{cases} \quad (3)$$

Assuming a fixed length  $L$  for fragments, the Entropy of the Markov distribution is given by

$$\mathcal{H}(M) = \frac{1}{L} \mathcal{H}(P_1) + \frac{L-1}{L} \sum_{q, q' \neq 0} \Pr(q \rightarrow q') \log \left( \frac{1}{\Pr(q \rightarrow q')} \right) \quad (4)$$

Empirical calculations show the entropy to be 3.3 bits. To match this, we use a custom encoding scheme (denoted as *Markov-encoding*) in which every transition  $q \rightarrow q'$  is encoded using a Huffman code of  $-\log(\Pr(q \rightarrow q'))$  bits. Table 1 summarizes the results of  $Q$ -value compression. The Markov encoding scheme provides a  $2.32 \times$  compression, requiring 3.45 bits per character. Further compression using bzip2 does not improve on this.

**Table 1.** Quality value compression results

	Raw File	bzip2	$\Delta$ (Huffman)	Markov (Huffman)
Bits per character	8	3.8	4.25	3.45
c.f.	1	2.11	1.88	2.32

## 5 Lossy Compression of Quality Values

Certainly, further compression of Quality values remains an intriguing research question. However, even with increasing sophistication, it is likely that  $Q$ -value compression will be the bottleneck in fragment-level compression. So we ask the sacrilegious question: *can the quality values be discarded?* Possibly in the future, base-calling will improve to the point that  $Q$ -values become largely irrelevant. Unfortunately, the answer today seems to be ‘no’. Many downstream applications including alignment, variant calling, and many others consider  $Q$ -values as a critical part of inference, and indeed, would not accept fragment data without  $Q$ -values. Here, we ask a different question: *is the downstream application robust to small changes in  $Q$ -values?* If so, a ‘lossy encoding’ could be immaterial to the downstream application.

Denote the number of distinct quality values produced as  $|Q| = Q_{\max} - Q_{\min}$ , which is encoded using  $\log_2(|Q|)$  bits. Note that a  $Q$ -score computation such as  $\lfloor -10 \cdot \log_2(P_{\text{err}}) \rfloor$  already involves a loss of precision. The error here can be reduced by rounding, or even better, by a ‘randomized’ rounding, defined as

$$\text{rrand}(x) = \begin{cases} \lceil x \rceil & \text{with probability } x - \lfloor x \rfloor \\ \lfloor x \rfloor & \text{otherwise} \end{cases} \quad (5)$$

Randomized rounding helps to prevent errors in subsequent interpretation. For example, suppose  $x = 1.4$  consistently over many experiments. Then, randomized rounding (which rounds  $x$  to 1 or 2) ensures that the expected value of  $\text{rrand}(x)$  is 1.4. For parameter  $b$ , we define the lossy Q-score encoding by

$$\text{LQ-score}_b(Q) = \text{rrand}\left(\frac{Q\text{-score} \cdot 2^b}{|Q|}\right) \quad (6)$$

We encode the  $2^b$  distinct values using Markov-encoding. A downstream application will therefore see  $Q_{\min} + |Q| \cdot \text{LQ-score}_b(Q)$  instead of the original value  $Q$ .

We test the impact of the lossy scheme on Illumina's downstream application called CASAVA [3] that calls alleles based on fragments and associated Q-scores. CASAVA was run over a 50M wide portion of the Chr 2 of GAHUM using the original  $Q$ -values, and it returned a set  $S$  of  $|S| = 17,021$  variants that matched an internal threshold (the allele quality must exceed 10; in heterozygous cases the respective threshold for the weak allele is 6). For each choice of parameter  $b \in \{1, \dots, 5\}$ , we reran CASAVA after replacing the original score  $Q$  with  $Q_{\min} + |Q| \cdot \text{LQ-score}_b(Q)$ . Denote each of the resulting variant sets as  $S_b$ . A variant  $s \in S \cap S_b$  is concordant. It is considered *discrepant* if  $s \in (S \setminus S_b) \cup (S_b \setminus S)$ .

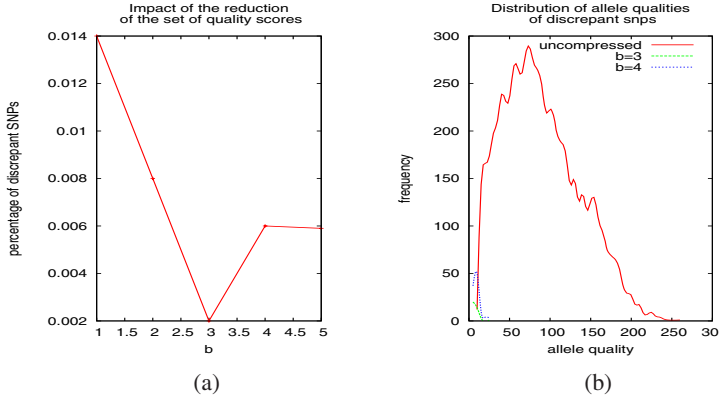
The results in Figure 4 are surprising. Even with  $b = 1$  (using 1 bit to encode  $Q$  values), 98.6% of the variant calls in  $S$  are concordant. This improves to 99.4% using  $b = 3$ . Moreover, we observe (in Fig. 4b) that the discrepant SNPs are close to the threshold. Specifically, 85% of the discrepant SNPs have allele qualities  $\leq 10$ .

## 5.1 Is the Loss-Less Scheme Always Better?

We consider the 38 positions in Chr 2 where the lossy (3-bits) compression is discrepant from the loss-less case. On the face of it, this implies a 0.2% error in SNP calling, clearly unacceptable when scaled to the size of the human genome. However, this assumes that the loss-less call is always correct. We show that this is clearly not true by comparing the SNP calls based on lossy and loss-less data in these 38 positions with the corresponding entries, if any, in dbSNP (version 29). We show that most discrepancies come from marginal decisions between homozygote and heterozygote calls.

For simplicity, we only describe the 26/38 SNPs with single nucleotide substitution in dbSNP. In all discrepant cases, the coverage is no more than 5 reads (despite the fact that the mean coverage is  $30\times$ ). Further, in all but 2 cases, both lossy, and lossless agree with dbSNP, and the main discrepancy is in calling heterozygote versus homozygotes. Specifically, lossy calls 14/10 homozygotes and heterozygotes, against lossless (12/12). With coverage  $\leq 5$ , the distinction between homozygote and heterozygotes is hard to make. Close to 50% of the differences were due to consideration of extra alleles due to lossy compression, while in the remaining, alleles are discarded. Given those numbers, it is totally unclear that our lossy compression scheme yields *worse* results than the lossless set, not to mention that in some cases it can lead to better results.

We next consider the two positions where the discrepant SNPs produced by the lossy scheme completely disagree with the dbSNP call. Table 2 shows that at position 43150343 dbSNP reports C/T. The loss-less  $Q$ -values and allele calls were respectively



**Fig. 4. Impact of Lossy Compression on CASAVA.** CASAVA was run on a 50M wide region of Chr 2 of GAHUM using lossless and lossy compression schemes. The  $y$ -axis plots the fraction of discrepant SNPs as a function of lossy compression. The  $x$ -axis shows the number of bits used to encode Q-scores. (b) The allele quality distribution of all lossless SNPs and the discrepant SNPs for 3 and 4-bit quantization. The plot indicates that the discrepant variants are close to the threshold of detection (allele quality of 6 for weak alleles in the heterozygous case, 10 for the homozygous case).

39G, 28G, 20G, 30G; CASAVA did not make a call. On the other hand, the lossy reconstruction led to values 41G, 27G, 22G, 32G, which pushed the overall allele quality marginally over the threshold, and led to the CASAVA call of ‘G’. In this case, the lossy reconstruction is quite reasonable, and there is no way to conclude that an error was made. The second discrepant case tells an identical story.

Given the inherent errors in SNP calling (lossy *or* lossless), we suggest that the applications of these SNP calls are inherently robust to errors. The downstream applications are usually one of two types. In the first case, the genotype of the individual is important in determining correlations with a phenotype. In such cases, small coverage of an important SNP must always be validated by targeted sequencing. In the second case, the SNP calls are used to determine allelic frequencies and SNP discovery in a population. In such cases, marginally increasing the population size will correct errors in individual SNP calls (especially ones due to low coverage). Our results suggest that we can tremendously reduce storage while not impacting downstream applications by coarsely quantizing quality values.

**Table 2. Case of wrongly called alleles.** In both cases the lossy quality values result in a score which marginally exceeds the threshold of 10 used to call the allele.

position	dbSNP entry	scheme	Qvalues				allele quality	Decision
43150343	C/T	lossy-8	41G	27G	22G	32G	10.2	G
		lossless	39G	28G	20G	30G	9.9	-
46014280	A/G	lossy-8	27C	37C	37C		10.1	C
		lossless	27C	36C	36C		9.9	-

## 6 Putting It All Together: Compression Results

We used SLIMGENE to compress the GAHUM data-set with 1.1B reads, a total size of 285GB. We focus on the columns containing the reads, their chromosome locations and match descriptors (124.7GB), and the column containing  $Q$ -values (103.4GB), for a total size of 228.1GB. The results are presented in Table 3 and show a 40 $\times$  compression of fragments. Using a lossy 1-bit encoding of  $Q$ -values results in a net compression of 14 $\times$  (8 $\times$  with a 3-bit encoding). While space restriction preclude a detailed comparison with other data representation formats like SAM/BAM, we report that the BAM representation of GAHUM results only in a 3 $\times$  compression of the dataset.

**Table 3. Compression of GAHUM using SLIMGENE.** Using a loss-less  $Q$ -value compression, we reduce the size by 5 $\times$ . A lossy  $Q$ -value quantization results in a further 3 $\times$  compression, with minimal effect on downstream applications.

	fragments+ alignment(GB)	$Q$ -values (GB)	total (GB)	execution time(hr)
Uncompressed	124.7	103.4	228.1	N/A
gzip (in isolation)	15.83	49.92	65.75	N/A
bzip2 (in isolation)	17.9	46.49	64.39	10.79
SLIMGENE	3.2	42.23	45.43	7.38
SLIMGENE+bzip2	3.04	42.34	45.38	7.38
SLIMGENE+lossy $Q$ -values( $b = 3$ )	3.2	26	29.8	7.38
SLIMGENE+lossy $Q$ -values( $b = 1$ )	3.2	13.5	16.7	7.38

## 7 Discussion

The SLIMGENE toolkit described here is available on request from the authors. While we have obtained compression factors of 35 or more for fragment compression, we believe we could do somewhat better and get closer to information theoretic limits. Currently, error-encoding is the bottleneck, and we do not distinguish between sequencing errors and genetic variation. By storing multiple (even synthetic) references, common genetic variation can be captured by exact matches instead of error-encoding. To do this, we only have to increase the POS vector while greatly reducing the number of ErrorInstructions. This trade-off between extra storage at the compressor/decompressor versus reduced transmission can be explored further.

While this paper has focused on fragment compression as opposed to sequence compression (Brandon et al. [2]), we believe both forms of compression are important, and in fact, complementary. In the *future*, if individuals have complete diploid genome sequences available as part of their personal health records, the focus will shift to sequence-level compression. It seems likely that fragment level compression will continue to be important to advance knowledge of human genetic variation, and is the pressing problem faced by researchers *today*. We note that Brandon et al. [2] also mention fragment compression briefly, but describe no techniques.

While we have shown 2-3 $\times$  compression of quality values, we believe it is unlikely this can be improved further. It is barely conceivable that unsuspected relations exist, which allow us to predict  $Q$ -values at some positions using  $Q$ -values from other positions; this can then be exploited for additional compression. However, there is nothing in the physics of the sequencing process that suggests such complicated correlations exist. Further, it would be computationally hard to search for such relations.

If compressing quality values beyond 3 $\times$  is indeed infeasible, then lossy compression is the only alternative for order of magnitude reductions. Our results suggest that the loss is not significant for interpretation. However, we have only scratched the surface. Using *companding* (from Pulse Code Modulation [11]), we plan to deviate from uniform quantization, and focus on wider quantization spacings for the middle quality values and smaller spacing for very high and very low quantization values. Further, we need to investigate the effect of quantization on other analysis programs for say *de novo* assembly, structural variation, and CNV detection. The number of quantization values in SLIMGENE is parameterized, and so different application programs can choose the level of quantization for their needs. A more intriguing idea is to use multi-level encoding as has been suggested for video [19]; initially, coarsely quantized quality values are transmitted, and the analysis program only requests finely quantized values if needed.

As sequencing of individuals becomes commoditized, its production will shift from large sequencing centers to small, distributed laboratories. Further, analysis is also likely to be distributed among specialists who focus on specific aspects of human biology. Our paper initiates a study of fragment compression, both loss-less and lossy, which should reduce the effort of distributing and synthesizing this vast genomic resource.

## Acknowledgements

VB and CK were supported by grants from the NSF (-III #0810905), and NIH (HG004962).

## References

1. Bellamy, J.C.: Digital Telephony, vol. 3rd. Wiley, Chichester (2000)
2. Brandon, M.C., Wallace, D.C., Baldi, P.: Data structures and compression algorithms for genomic sequence data. *Bioinformatics* 25, 1731–1738 (2009)
3. The CASAVA software toolkit, <http://www.illumina.com/pages.ilmn?ID=314>
4. Chen, X., Li, M., Ma, B., Tromp, J.: DNACompress: fast and effective DNA sequence compression. *Bioinformatics* 18, 1696–1698 (2002)
5. Christley, S., Lu, Y., Li, C., Xie, X.: Human genomes as email attachments. *Bioinformatics* 25, 274–275 (2009)
6. Dublin, M.: So Long, Data Depression (2009), <http://www.genomeweb.com/informatics/so-long-data-depression>
7. Feuk, L., Carson, A.R., Scherer, S.W.: Structural variation in the human genome. *Nat. Rev. Genet.* 7(2), 85–97 (2006)
8. Helicos Biosciences, <http://www.helicosbio.com/>
9. Iafrate, A.J., Feuk, L., Rivera, M.N., Listewnik, M.L., Donahoe, P.K., Qi, Y., Scherer, S.W., Lee, C.: Detection of large-scale variation in the human genome. *Nat. Genet.* 36(9), 949–951 (2004)

10. The Illumina Genome Analyzer, <http://www.illumina.com/sequencing/>
11. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R.: The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 2078–2079 (2009)
12. Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., Zhang, H.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics* 17, 149–154 (2001)
13. <http://www.mpeg.org>
14. Newman, T.L., Tuzun, E., Morrison, V.A., Hayden, K.E., Ventura, M., McGrath, S.D., Rocchi, M., Eichler, E.E.: A genome-wide survey of structural variation between human and chimpanzee. *Genome Res.* 15(10), 1344–1356 (2005)
15. Pacific BioSciences, <http://www.pacificbiosciences.com/index.php>
16. Roche 454 Sequencing, <http://www.454.com/>
17. The SAM/BAM format, <http://samtools.sourceforge.net/SAM1.pdf>
18. Sharp, A.J., Cheng, Z., Eichler, E.E.: Structural variation of the human genome. *Annu. Rev. Genomics Hum. Genet.* (June 2006)
19. Steven, E.A., Mccanne, S., Vetterli, E.: A Layered Dct Coder For Internet Video. In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 13–16 (1996)
20. Ziv, J., Lempel, A.: Compression of Individual Sequences Via Variable-Rate Coding. *IEEE Transactions on Information Theory* (1978)

# On the Genealogy of Asexual Diploids

Fumei Lam<sup>1</sup>, Charles H. Langley<sup>2</sup>, and Yun S. Song<sup>3,4</sup>

<sup>1</sup> Department of Computer Science, University of California, Davis, CA 95616, USA

<sup>2</sup> Section of Evolution and Ecology, University of California, Davis, CA 95616, USA

<sup>3</sup> Computer Science Division, University of California, Berkeley, CA 94720, USA

<sup>4</sup> Department of Statistics, University of California, Berkeley, CA 94720, USA

flam@cs.ucdavis.edu, chlangley@ucdavis.edu, yss@eecs.berkeley.edu

**Abstract.** Given molecular genetic data from diploid individuals that, at present, reproduce mostly or exclusively asexually without recombination, an important problem in evolutionary biology is detecting evidence of past sexual reproduction (i.e., meiosis and mating) and recombination (both meiotic and mitotic). However, currently there is a lack of computational tools for carrying out such a study. In this paper, we formulate a new problem of reconstructing diploid genealogies under the assumption of no sexual reproduction or recombination, with the ultimate goal being to devise genealogy-based tools for testing deviation from these assumptions. We first consider the infinite-sites model of mutation and develop linear-time algorithms to test the existence of an asexual diploid genealogy compatible with the infinite-sites model of mutation, and to construct one if it exists. Then, we relax the infinite-sites assumption and develop an integer linear programming formulation to reconstruct asexual diploid genealogies with the minimum number of homoplasy (back or recurrent mutation) events. We apply our algorithms on simulated data sets with sizes of biological interest.

## 1 Introduction

Reproduction in asexual organisms usually is less costly than that in sexual organisms. Yet, sexual reproduction and genetic recombination are common to the majority of higher organisms in nature, and several different explanations have been put forward to address this intriguing phenomenon (see [4, 21] and references therein). Although it still remains debatable as to which precise evolutionary conditions and mechanisms maintain sex and recombination in natural populations, it is widely believed that sex and recombination are important for the long-term evolutionary success of an organism; that is, asexual organisms are believed to be much more susceptible to extinction than are their sexual counterparts that undergo meiosis and mating [25]. Contrary to this common belief, the phylum Rotifera, microscopic aquatic animals widespread throughout the world, contains a class—namely, Bdelloidea—that seems to have been reproducing asexually for tens of millions of years, diversifying into 360 known species that constitute 4 families and 18 genera. Fossil evidence suggests that bdelloid rotifers have been around for at least 35 to 40 million years [34], while

molecular genetic analysis suggests an age that is more than twice as large [23]. Maynard Smith [26] referred to the bdelloid rotifers as “something of an evolutionary scandal”, and it has been questioned in the past whether they indeed have remained asexual for all that while [19].

Recently, Mark Welch and Meselson [23] analyzed molecular genetic data of four bdelloid species and provided evidence to support bdelloid rotifers’ ancient, continuous asexuality. Their method was based on counting synonymous sequence differences between different copies of a gene within individual, which, under neutrality, are expected to be over-represented in an old asexual organism. (See [5] for a review.) Mark Welch and Meselson showed that allelic sequence differences at synonymous sites are significantly greater in bdelloid rotifers than in their closest relative class monogonont rotifers, consisting of about 1500 species, which seem to reproduce mostly asexually, but with an occasional sexual reproduction. (More recent evidence in support of the ancient asexuality of bdelloid rotifers is provided in [10].) In contrast to this success, when a similar analysis was applied to other asexual organisms such as darwinulid ostracods [28], of which morphological evidence strongly supports ancient asexuality [24], no significantly high level of sequence divergence was observed. In another study, a similar sequence divergence test applied to plant-parasitic worms (specifically, root-knot nematodes from the genus *Meloidogyne*) supported their ancient asexuality, while further analysis revealed that interspecific hybridization was involved in the history of this group [22]. From this study, the author concluded “genetic signatures of ancient asexuality must be taken with caution due to the confounding effect of interspecific hybridization, which has long been implicated in the origins of apomictic species.” As these cases illustrate, a more refined method that makes better use of DNA data is needed for studying asexuality.

In this paper, we develop new methods to test asexuality by explicitly considering the evolutionary history of diploid individuals. We first consider the infinite-sites model of mutation, which corresponds to the ideal case in which mutations provide as much information about genealogy as possible. This ideal case should provide an upper bound on our chance of detecting signatures of past sexual reproduction. Given  $n$  pairs of phased haplotypes or  $n$  unphased genotypes, our goal is to test the existence of an  $n$ -leaved *diploid perfect phylogeny* (DPP)—an asexual diploid genealogy compatible with the infinite-sites model of mutation and no recombination—for the input individuals, and to construct one if it exists. We devise linear-time algorithms for both phased haplotypic and unphased genotypic input data, and show that a minimal DPP for a given data set is unique if it exists. If a DPP solution exists for unphased genotypic input data, our algorithm finds a phasing of the input genotypes into pairs of haplotypes compatible with the DPP, and the DPP serves as a data structure that encodes all such phasing solutions.

In the second part of this paper, we relax the infinite-sites assumption and study the *diploid imperfect phylogeny* (DIP) problem, which is to reconstruct asexual diploid genealogies with the minimum number of homoplasmy (recurrent or back mutation) events. If the minimum number of homoplasmy events is



significantly greater than that expected for typical asexual organisms, then it may indicate that other evolutionary forces such as recombination, hybridization, or sexual reproduction may have played a role in the evolutionary history. We develop an integer linear programming formulation to tackle this problem and study the practicality of our approach by applying our algorithms on simulated data sets with sizes of current biological interest.

Our ultimate goal is to devise genealogy-based tools for testing deviation from asexual evolution. Given molecular genetic data from diploid individuals that, at present, reproduce mostly or exclusively asexually, an important open problem is to estimate the frequency of past sexual reproduction, as well as the amount of recombination (meiotic and mitotic crossovers and gene-conversions). Further, it will be important to estimate when sexuality was lost and how many independent times. The work described in this paper is a modest step toward that general direction. The preliminary results described here suggest that genealogical approaches may provide new insights into the study of asexual evolution.

CLONETREE, software that implements our algorithms, will be made publicly at <http://www.eecs.berkeley.edu/~yss/software.html>. It produces a graphical output that displays the diploid genealogy found by our algorithms.

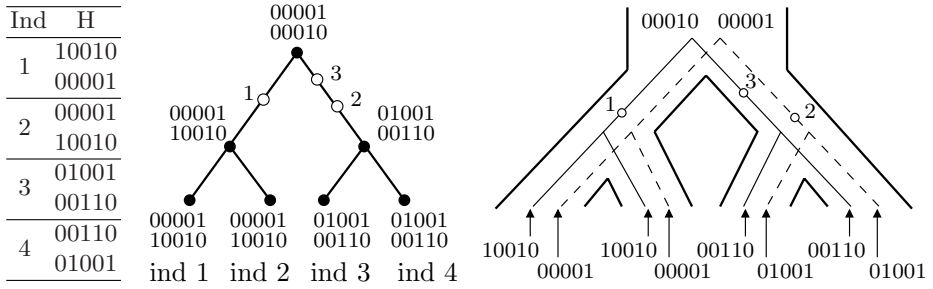
## 2 Diploid Perfect Phylogeny

We assume that the input data consist of either phased or unphased single nucleotide polymorphisms (SNPs) from  $n$  diploid individuals with  $m$  polymorphic sites. Each site has at most two phased alleles, denoted by  $\{0, 1\}$ . The data we consider are of the following two types:

**Definition 1 (Haplotype data).** *A haplotype is a binary string of length  $m$ . Let  $h_i$  and  $\tilde{h}_i$  denote the pair of haplotypes of individual  $i$ ;  $h_i$  and  $\tilde{h}_i$  are called mates. A collection of such pairs of haplotypes for  $n$  individuals is denoted by a  $2n$ -by- $m$  binary matrix  $H$ , in which rows  $2i - 1$  and  $2i$  correspond to the two haplotypes of individual  $i$ .*

**Definition 2 (Genotype data).** *Let  $g_i$  denote the genotype of individual  $i$ . The value of  $g_i$  at site  $k$  is 0 if individual  $i$  has two copies of 0 at site  $k$ ; 1 if individual  $i$  has two copies of 1 at site  $k$ ; or 2 otherwise. A collection of genotypes for  $n$  individuals is denoted by  $G$ , with row  $i$  corresponding to  $g_i$ . A  $2n$ -by- $m$  binary matrix  $H$  is said to be a phasing solution to an  $n$ -by- $m$  ternary matrix  $G$  if, for all  $i = 1, \dots, n$ ,  $g_i$  in  $G$  is the genotype consistent with the mates  $h_i$  and  $\tilde{h}_i$  in  $H$ .*

In the infinite-sites model of mutation, at most one mutation may occur per site in the entire evolutionary history. Trees representing evolutionary histories consistent with the infinite-sites model of mutation are called perfect



**Fig. 1.** From left to right, a haplotype data set  $H$  for four individuals, its unique minimal diploid perfect phylogeny  $T$ , and evolutionary histories of the haplotypes embedded in  $T$ . We use  $\tau_X$  and  $\tau_Y$  to denote the solid and the dotted trees in  $T$ , respectively. An open circle labeled  $k$  represents a mutation at site  $k$ .

phylogenies [29]. We will refer to these as *haploid perfect phylogenies (HPP)* to distinguish them from *diploid perfect phylogenies*, a new concept defined as follows.

**Definition 3 (Diploid Perfect Phylogeny).** A diploid perfect phylogeny (DPP) for  $n$  diploid individuals is an  $n$ -leaved rooted tree  $T$  representing the evolutionary history of self-cloning (or asexually reproducing) individuals satisfying:

1. Mutations occur on edges and each site may mutate at most once in  $T$ . Time flows from the root (which has degree 2) to the leaves (which have degree 1), and each edge in  $T$  represents a diploid lineage. If site  $k$  mutates on an edge, only one of the two haplotypes gets modified at that site, and the newly arising allele (0 or 1) has never been seen before at that site.
2. Depending on whether the input data are pairs of haplotypes or genotypes, every vertex of a DPP is labeled by a pair of haplotypes or a genotype, respectively.
3. There is a one-to-one correspondence between the  $n$  leaves of  $T$  and the  $n$  input individuals.

A minimal DPP is a DPP in which the two ends of every interior edge have different labels.

Note that a set of  $2n$  haplotypes for  $n$  individuals may admit an HPP solution while admitting no DPP solution. A DPP example is shown in the middle of Figure 1. In this paper, we address the following two algorithmic questions:

**DPP for Haplotype Data:** Given a haplotypic data set  $H$  for  $n$  diploid individuals, determine whether  $H$  admits a DPP solution, and find one if it exists.

**DPP Haplotyping for Genotype Data:** Given a genotypic data set  $G$  for  $n$  diploid individuals, determine whether  $G$  can be phased to a haplotypic data set  $H$  that admits a DPP solution, and if so, find such a phasing solution  $H$ .

### 3 DPP for Haplotype Data

In [13], Gusfield devised a linear-time algorithm to test whether a given haplotypic input data set admits an HPP solution and to find one if it exists. In this section, we construct an analogous linear-time algorithm for DPP, making use of Gusfield’s linear-time algorithm for HPP. First, we highlight several important properties satisfied by DPPs.

#### 3.1 Properties of Diploid Perfect Phylogenies

Suppose there is an  $n$ -leaved minimal DPP  $T$  for  $H$ . Let  $x_r$  and  $y_r$  denote the root haplotypes of  $T$ . Following the history of  $x_r$  on  $T$  leads to one haplotype per leaf in  $T$ . Denote this set of haplotypes  $H_X$  and their history  $\tau_X$ . Similarly, follow the history of  $y_r$  on  $T$  to obtain  $H_Y$  and  $\tau_Y$ . Note that each diploid individual has exactly one of its two haplotypes in  $H_X$  and the other in  $H_Y$ . The following properties are implied by the one-mutation-per-site condition:

- P1. The set of mutations in  $\tau_X$  and  $\tau_Y$  are disjoint. (In Figure 1, sites 1 and 3 mutate in  $\tau_X$  but not in  $\tau_Y$ . Similarly, site 2 mutates in  $\tau_Y$  but not in  $\tau_X$ .)
- P2. If  $x_r[k] \neq y_r[k]$ , then both 0 and 1 have already been seen, so part 1 of Definition 3 implies that neither  $\tau_X$  nor  $\tau_Y$  contains a mutation at site  $k$ . As a consequence, no individual in  $T$  is homozygous at site  $k$ . (In Figure 1, sites 4 and 5 satisfy this property.)

For a given input data set  $H$ , the one-mutation-per-site condition imposes tight constraints on the possible root haplotypes of a DPP. In what follows, we use  $\mathcal{E}(H)$  to denote the set of all sites in  $H$  at which every individual is heterozygous.

**Lemma 1 (Constraints on the root).** *The haplotypes  $x_r, y_r$  of any possible root individual of a DPP satisfy the following properties:*

1. For all  $k \notin \mathcal{E}(H)$ , there cannot be two distinct homozygous genotypes at site  $k$ . If any individual  $i$  in  $H$  has a homozygous genotype  $h_i[k] = \tilde{h}_i[k] = c$ , then the root individual also has the same genotype  $x_r[k] = y_r[k] = c$ .
2.  $H$  restricted to the sites in  $\mathcal{E}(H)$  has exactly two distinct haplotypes, and those haplotypes are equal to the root haplotypes  $x_r, y_r$  restricted to  $\mathcal{E}(H)$ . More precisely, for any particular site  $j \in \mathcal{E}(H)$ , let  $H_X$  (respectively,  $H_Y$ ) denote the set of  $n$  haplotypes with a 1 (respectively, 0) at site  $j$ . Then, for all  $k \in \mathcal{E}(H)$ , both  $H_X$  and  $H_Y$  are non-polymorphic at site  $k$ , with  $H_X$  and  $H_Y$  having different alleles. Further,  $x_r$  (respectively,  $y_r$ ) restricted to the sites in  $\mathcal{E}(H)$  is the same as any haplotype in  $H_X$  (respectively,  $H_Y$ ) restricted to  $\mathcal{E}(H)$ . So, for all  $k \in \mathcal{E}(H)$ , the root is heterozygous at site  $k$ .

*Proof.* If there exists a DPP, Property P1 implies that no two distinct homozygous genotypes may exist any at site. Further, Property P2 implies that if  $H$  contains an individual homozygous at site  $k$ , then the root individual of any

DPP solution for  $H$  must be homozygous at that site. The first part of this lemma then follows from these two facts.

Let  $T$  denote an  $n$ -leaved DPP for the  $n$  individuals in  $H$ , and suppose that the root  $\rho$  of  $T$  is homozygous at some site  $j \in \mathcal{E}(H)$ . Then, since every individual in  $H$  is heterozygous at that site,  $\rho$  is not in  $H$ . Now, the one-mutation-per-site condition implies that there is an edge that separates  $\rho$  from all individuals in  $H$ , thus implying that  $T$  contains a leaf not labeled by any individual in  $H$ , which in turn implies that  $T$  is not an  $n$ -leaved DPP for  $H$ , a contradiction. Hence, the root individual of a DPP must be heterozygous at every site  $j \in \mathcal{E}(H)$ . This fact and the one-mutation-per-site condition together imply that there is no mutation event at any site  $j \in \mathcal{E}(H)$  in the entire  $T$ , and the second part of the lemma immediately follows.  $\square$

Lemma [□](#) implies that if a DPP exists for  $H$ , there is a unique choice for the root. Using this lemma, we can show several useful results that hold if a DPP exists. First, we need two definitions.

**Definition 4 (Resolution of a vertex).** *In a graph  $\mathcal{G}$ , resolution of a degree- $d$  vertex  $v$  incident to edges  $e_1, \dots, e_d$  (with  $d > 3$ ), is an operation that splits  $v$  into two new vertices  $v_1$  and  $v_2$ , such that (i)  $v_1$  and  $v_2$  are joined by a new edge, (ii) each of  $e_1, \dots, e_d$  is incident with either  $v_1$  or  $v_2$ , (iii) both  $v_1$  and  $v_2$  have degree  $\geq 3$ , and (iv) the remaining vertices and edges of  $\mathcal{G}$  remain the same.*

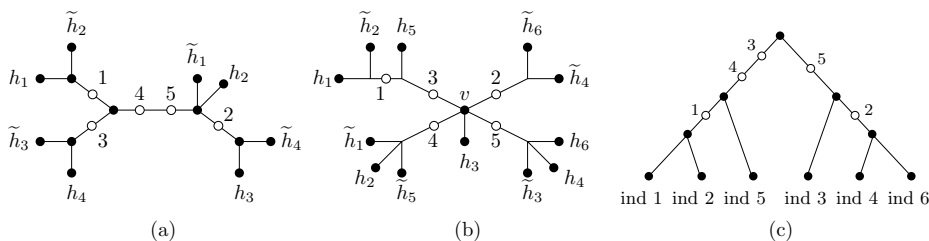
**Definition 5 ( $\bowtie$ , Join operation).** *For two  $k$ -by- $l$  matrices  $M_1$  and  $M_2$ , the  $k$ -by- $2l$  matrix  $M_1 \bowtie M_2$  is obtained by appending row  $i$  of  $M_2$  to row  $i$  of  $M_1$ .*

The following result provides a way to find the partition of  $H$  into  $H_X$  and  $H_Y$  if a DPP solution exists.

**Proposition 1 (Partition of  $H$  into  $H_X$  and  $H_Y$ ).** *For  $n > 1$ , suppose there exists an  $n$ -leaved diploid perfect phylogeny  $T$  for the  $n$  individuals in  $H$ . Then, the  $2n$  haplotypes in  $H$  admit a unique  $2n$ -leaved minimal unrooted haploid perfect phylogeny  $\tau$  satisfying the following:*

1. *If  $\mathcal{E}(H) \neq \emptyset$ , then there exists a unique edge in  $\tau$  such that cutting that edge partitions  $\tau$  into two  $n$ -leaved subtrees such that, for each individual  $i$ , haplotype  $h_i$  appears as a leaf of one subtree while its mate haplotype  $\tilde{h}_i$  appears as a leaf of the other subtree. (See [Figure 2a](#).)*
2. *If  $\mathcal{E}(H) = \emptyset$ , then there exists a unique vertex  $v$  in  $\tau$  with degree  $d$ , where  $d > 3$ , such that resolving  $v$  and cutting the edge between the newly created vertices partitions  $\tau$  into two  $n$ -leaved subtrees such that, for each individual  $i$ , haplotype  $h_i$  appears as a leaf of one subtree while its mate haplotype  $\tilde{h}_i$  appears as a leaf of the other subtree. (See [Figure 2b](#).)*

*Proof.* Define  $\tau_X, \tau_Y, x_r$ , and  $y_r$  as in the beginning of this section. If  $x_r$  and  $y_r$  are not identical, then add a new edge between the root of  $\tau_X$  and the root of  $\tau_Y$ , and add mutation events on that edge for all sites  $k$  where  $x_r[k] \neq y_r[k]$ . If  $x_r$  and  $y_r$  are identical, combine  $\tau_X$  and  $\tau_Y$  by identifying the root vertex  $\rho_X$  of



**Fig. 2.** HPP and DPP examples for Propositions 1 and 2. (a) There is a unique edge (namely, the edge on which sites 4 and 5 mutate) satisfying the property described in part 1 of Proposition 1. (b) There is a unique vertex (labeled  $v$  in the figure) satisfying the property described in part 2 of Proposition 1. (c) The unique minimal diploid perfect phylogeny consistent with the HPP shown in (b).

$\tau_X$  with the root vertex  $\rho_Y$  of  $\tau_Y$ , such that the new vertex  $\rho_{X \oplus Y}$  obtained from identifying  $\rho_X$  and  $\rho_Y$  is incident with all the edges that were incident with  $\rho_X$  or  $\rho_Y$ . Then, Properties P1 and P2 implies that the resulting tree  $\tau_{X \oplus Y}$  is an unrooted HPP for the haplotypes in  $H$ . Now, contract interior edges in  $\tau_{X \oplus Y}$  with no mutations and call the resulting tree  $\tau$ . Note that  $\tau$  is a unique  $2n$ -leaved minimal unrooted HPP for the haplotypes in  $H$ .

If  $\mathcal{E}(H) \neq \emptyset$ , then  $x_r$  and  $y_r$  are not identical, so  $\tau$  described above contains an edge between  $x_r$  and  $y_r$  with at least one mutation. Therefore,  $\tau$  satisfies part 1 of the Proposition. If  $\mathcal{E}(H) = \emptyset$ , then Lemma 1 implies that the root haplotypes  $x_r$  and  $y_r$  are identical, so  $\tau$  has the vertex  $\rho_{X \oplus Y}$  described above. Note that  $\rho_{X \oplus Y}$  has degree  $> 3$ , and part 2 of the Proposition is satisfied by construction.  $\square$

If a DPP exists for  $H$  and  $\mathcal{E}(H) = \emptyset$ , part 2 of Proposition 1 indicates that there is a unique degree- $d$  vertex  $v$ , where  $d > 3$ , such that a partition of  $H$  into  $H_X$  and  $H_Y$  can be obtained from resolving that vertex. However, there can be more than one admissible resolution of that vertex (and hence more than one possible partition of  $H$ ) that is consistent with the existence of a DPP. Below, we show that all such resolutions imply the same minimal DPP. For illustration, consider the HPP shown in Figure 2b. It admits two possible partitions of  $H$  into  $H_X$  and  $H_Y$ —either  $H_X = \{h_1, h_2, h_3, \tilde{h}_4, h_5, \tilde{h}_6\}$  and  $H_Y = \{\tilde{h}_1, h_2, \tilde{h}_3, h_4, \tilde{h}_5, h_6\}$ , or  $H_X = \{h_1, \tilde{h}_2, \tilde{h}_3, h_4, h_5, h_6\}$  and  $H_Y = \{\tilde{h}_1, h_2, h_3, \tilde{h}_4, \tilde{h}_5, \tilde{h}_6\}$ . It is easy to see that both cases lead to the same DPP, depicted in Figure 2c. Now, the following result establishes the uniqueness of a minimal DPP solution:

**Proposition 2 (Uniqueness).** *If a DPP exists for  $H$ , then  $H$  admits a unique minimal DPP.*

*Proof.* Suppose that  $H$  admits a DPP. If  $\mathcal{E}(H) \neq \emptyset$ , then part 1 of Proposition 1 implies that there is a unique way to partition  $H$  into  $H_X$  and  $H_Y$ . The root haplotypes  $x_r$  and  $y_r$  are as described in Lemma 1. A minimal DPP for  $H$  must

be a minimal HPP for  $H_X \bowtie H_Y$  with  $x_r \bowtie y_r$  as the root sequence, and its uniqueness follows from the uniqueness of a rooted minimal HPP for a binary matrix with a given root.

Suppose that  $\mathcal{E}(H) = \emptyset$ , and let  $\tau$  and  $v$  be as in Proposition [1](#). Lemma [1](#) implies that the root haplotypes  $x_r$  and  $y_r$  of a DPP are identical. For ease of exposition, suppose that those haplotypes are all-zero. Then, the haplotypes assigned to  $v$  are all-zero. To each mutation in  $\tau$ , one can associate a binary character for the  $n$  individuals  $\{1, 2, \dots, n\}$  as follows. For each mutation occurring on some edge in  $\tau$ , imagine cutting that edge, and consider the subtree not containing  $v$  that would be cut. Assign a 1 to every individual  $i$  with either  $h_i$  or  $\tilde{h}_i$  as a leaf in that subtree, and assign 0s to all other individuals. (Since there exists a DPP, no individual has both of its haplotypes in that subtree.) Now, the tree shape and the assignment of mutations to the edges of a minimal DPP for  $H$  must be the same as that of a minimal HPP for the set of binary characters just described with the all-zero sequence as the root, and the uniqueness of that DPP is immediate.  $\square$

### 3.2 A Linear-Time Algorithm for Haplotype Data

Using the above results and Gusfield's linear-time algorithm for HPP [13](#), we can devise the following  $O(mn)$ -time algorithm to find a DPP solution, if it exists:

1. Check that the conditions in Lemma [1](#) are satisfied. If not, there is no DPP solution. Otherwise, the root haplotypes  $x_r$  and  $y_r$  are uniquely determined.
2. Check whether there exists a  $2n$ -leaved minimal unrooted HPP  $\tau$  for  $H$ . If not, there is no DPP solution. Otherwise, check whether a partition of the  $2n$ -by- $m$  input matrix  $H$  into two  $n$ -by- $m$  matrices  $H_X$  and  $H_Y$  can be found as described in Proposition [1](#).
  - (a) If  $\mathcal{E}(H) \neq \emptyset$ , the two ends of the edge in  $\tau$  needed to be cut should be labeled by  $x_r$  and  $y_r$ .
  - (b) If  $\mathcal{E}(H) = \emptyset$ , then  $x_r = y_r$ . The vertex  $v$  described in the second part of Proposition [1](#) is the one labeled by  $x_r = y_r$ . Determine whether there exists a resolution of  $v$  such that cutting the newly created edge partitions  $H$  into  $H_X$  and  $H_Y$ . If not, there is no DPP solution.
3. Test whether there exists an  $n$ -leaved HPP for the  $n$ -by- $2m$  matrix  $H_X \bowtie H_Y$  with  $x_r \bowtie y_r$  as the root sequence. If so, then it corresponds to the unique minimal DPP for  $H$ . Otherwise, there is no DPP solution.

## 4 DPP Haplotyping for Genotype Data

In [14](#), Gusfield considered phasing (or haplotyping) genotypic input data as an HPP and provided a nearly-linear-time algorithm for the problem. Simpler but slower solutions [2,8](#) were subsequently proposed for the problem, and linear-time algorithms were recently found [7,33](#). The absence of recombination and

homoplasmy imposes stringent constraints on the genealogy of asexual diploid individuals. In this section, we exploit such constraints to devise a simple linear-time algorithm for the DPP Haplotyping Problem under the assumption of asexual reproduction. Our approach has two stages. First, for a given input genotype data set  $G$ , we find a DPP if it exists. Then, we use that DPP to find a phasing solution for  $G$ .

#### 4.1 A Linear-Time Algorithm for Constructing a DPP for Genotype Data

Lemma 1 implies that genotypic states 0 and 1 (denoting homozygotes) cannot both appear in any column in  $G$ . Further, the one-mutation-per-site condition implies that, when a mutation occurs at a site, it is either of type  $0 \rightarrow 2$  or  $1 \rightarrow 2$ , but never  $2 \rightarrow 0$  or  $2 \rightarrow 1$ . Using these facts, we devise the following linear-time algorithm for constructing a DPP for  $G$ , if it exists:

1. For every column  $k = 1, \dots, m$  in  $G$  do the following:
  - (a) Check if both 0 and 1 appear in column  $k$ . If so, there is no DPP solution.
  - (b) If column  $k$  contains neither a 0 nor a 1, then set  $z_r[k] = 2$ .
  - (c) Else, if column  $k$  contains a 0 (1), then set  $z_r[k] = 0$  ( $z_r[k] = 1$ ).
2. If the above step has not failed, then there are at most two distinct genotypic states in each column of  $G$ . Viewing each column as a two-state character and each row as a haplotype, test whether  $G$  admits an  $n$ -leaved HPP with  $z_r$  as the root sequence, with mutations of type  $0 \rightarrow 2$  or  $1 \rightarrow 2$ , depending on the root character state. If such an HPP exists, it corresponds to the unique minimal DPP for  $G$ .

With appropriate renaming of character states, the above algorithm can be carried out in  $O(mn)$  time using Gusfield's linear-time HPP algorithm for binary matrices [13]. Note that if a DPP exists for an input genotype data, its root genotype  $z_r$  is uniquely determined as described in the above algorithm.

Due to space considerations, we omit the details of our algorithm for finding a DPP haplotyping solution.

## 5 Diploid Imperfect Phylogeny

If a set of diploid sequences does not allow a diploid perfect phylogeny, then other forces must be present in the evolutionary history. These may include homoplasmy or recombination events and further analysis is necessary to distinguish between these possibilities.

**Definition 6 (Diploid Imperfect Phylogeny).** *A diploid imperfect phylogeny (DIP) for  $n$  diploid individuals is an  $n$ -leaved rooted tree  $T$  satisfying conditions (2) and (3) in the definition of Diploid Perfect Phylogeny, and satisfying condition (1) with the modification that multiple mutations are possible at each site.*

In order to measure the strength of evidence to distinguish between homoplasy and recombination events, we define a measure of deviation from a diploid perfect phylogeny. For a diploid imperfect phylogeny  $T$  displaying a set of sequences  $S$ , let  $M_T(k)$  denote the number of edges in  $T$  corresponding to mutation at site  $k$ .

**Definition 7.** A diploid imperfect phylogeny  $T$  for input  $H$  is  $q$ -imperfect (or  $q$ -near-perfect) if  $\sum_{k: M_T(k) \geq 1} (M_T(k) - 1) = q$ .

The diploid imperfect phylogeny problem is to find a DIP  $T$  displaying the input sequences which minimizes the imperfection  $q$ . In particular, if the sequences can be displayed in a diploid perfect phylogeny  $T$ , then  $M_T(k) \leq 1$  for each site  $k$ , and  $T$  satisfies  $q = 0$ .

In the case of haploid input sequences, the problem of constructing imperfect haploid phylogenies has received much attention from both theoretical and practical points of view. Fernandez-Baca and Lagergren [9], Halperin and Eskin [17], and Sridhar et al. [31] analyzed theoretical bounds for algorithms to solve this problem to optimality, while Sridhar et al. [30] showed that the problem is fixed-parameter tractable in the imperfection of the resulting phylogeny. Further, it has been shown that linear programming approaches can efficiently handle data sets of biological interest [32]. We now consider the case of constructing diploid imperfect phylogenies and introduce a problem which casts this problem in the framework of combinatorial optimization.

## 5.1 Group Steiner Tree Problem

The problem of reconstructing phylogenies is closely related to the *Steiner Tree Problem*, a well studied problem in combinatorial optimization. Given a graph  $G = (V, E)$  with edge costs and a set of terminals  $R \subseteq V$ , a *Steiner tree* in  $G$  is a subgraph of  $G$  containing a path between any pair of terminals. The cost of a Steiner tree  $T$  is the sum of the edge costs in  $T$  and the Steiner Tree Problem is to find the minimum cost Steiner tree in  $G$ .

Let  $H$  be a set of input sequences of length  $m$  and let graph  $G$  be the  $m$ -cube defined on vertices  $V = \{0, 1\}^m$  and edges  $E = \{(u, v) \in V \times V : \sum_i |u_i - v_i| = 1\}$ . Let  $R \subseteq V$  be the set of binary sequences corresponding to the rows of input  $H$ . The minimum (haploid) imperfect phylogeny problem is then equivalent to the minimum Steiner tree problem on underlying graph  $G$  with terminal vertices  $R$ . Even in this restricted setting, the Steiner tree problem is NP-complete [11].

To solve the *diploid* imperfect phylogeny problem, we introduce the following more general Steiner tree problem. Let  $G = (V, E)$  be an undirected graph, let  $d$  be a non-negative cost function on edge set  $E$ , and let  $R = R_1 \cup R_2 \dots R_k \subseteq V$  be a partition of the terminal vertices into disjoint groups. A *group Steiner tree* of  $G$  is a Steiner tree containing at least one vertex from each group  $R_i$  and the Group Minimum Steiner Tree (GMST) Problem is to find the group Steiner tree of minimum cost.

The diploid imperfect phylogeny problem can be transformed to an instance of the Group Steiner Tree problem as follows. Let  $H = \{h_i, \tilde{h}_i\}_{i=1}^n$  be the input



set of paired haplotype sequences to the diploid imperfect phylogeny problem, where  $h_i$  and  $\tilde{h}_i$  are binary sequences of length  $m$ . Let graph  $G$  be the  $2m$ -cube (where vertices are binary sequences of length  $2m$  and edges are pairs of binary sequences with Hamming distance equal to one), and for each  $i$ , let terminal group  $R_i$  be the pair of vertices  $\{h_i\tilde{h}_i, \tilde{h}_ih_i\} \subseteq V(G)$ . The GMST on this instance is then equivalent to the minimum diploid imperfect phylogeny problem on  $H$ .

Because of its computational complexity, an important component of any computational approach for solving the Steiner Tree problem is to eliminate vertices that cannot be present in *any* optimal tree. In the haploid imperfect phylogeny problem, it has been shown that the *Buneman graph* of the input sequences contains all optimal trees [3, 6]. Restricting the underlying graph of the problem in such a way has been shown to be efficient and practical on real data sets [32]. The following proposition shows an analogous results holds for the diploid phylogeny problem:

**Proposition 3.** *Let  $H$  be a set of  $n$  pairs of haplotype sequences  $\{h_i, \tilde{h}_i\}$  and let  $\mathcal{B}(H)$  denote the Buneman graph on  $\cup_i \{h_i\tilde{h}_i, \tilde{h}_ih_i\}$ . Then every minimum imperfect diploid phylogeny  $T^*(H)$  is a subgraph of  $\mathcal{B}(H)$ .*

We prove this proposition using the following theorem of Bandelt et al. for haploid imperfect phylogeny construction:

**Theorem 1 (Bandelt et al.).** [3, 29] *For binary haplotype input sequences  $H$ , let  $\mathcal{B}(H)$  denote the Buneman graph on  $H$ . Then every minimum imperfect phylogeny  $T^*(H)$  is a subgraph of  $\mathcal{B}(H)$ .*

*Proof (Proposition 3).* Let  $H = \{h_i, \tilde{h}_i\}_{i=1}^n$  be a set of  $n$  pairs of haplotype sequences of length  $m$ . Suppose  $T^*(H)$  is a minimum GMST on the hypercube of dimension  $2m$  with terminal groups  $R_i = \{h_i, \tilde{h}_i\}$  ( $1 \leq i \leq n$ ). By definition,  $T^*(H)$  must contain at least one terminal  $t_i$  from each terminal group  $R_i = \{h_i\tilde{h}_i, \tilde{h}_ih_i\}$ . It follows that  $T^*(H)$  is a minimum Steiner tree on terminal set  $\{t_i\}_{i=1}^n$ . By Theorem 1,  $T^*(H)$  is a subgraph of the Buneman graph  $\mathcal{B}(\{t_i\}_{i=1}^n)$ . Since  $t_i \in \{h_i\tilde{h}_i, \tilde{h}_ih_i\}$ , it follows that  $T^*(H)$  is a subgraph of the Buneman graph  $\mathcal{B}(\{h_i\tilde{h}_i, \tilde{h}_ih_i\}_{i=1}^n) = \mathcal{B}(H)$ . □

### 5.2 Integer Linear Programming Formulation

One approach for solving Steiner tree problems is to use integer linear programming (ILP) methods. We use the multicommodity flow formulation for the GMST problem, in which one unit of flow is sent from the root vertex to every group. For a subgraph  $S$  of a graph  $G$ , associate a vector  $x^S \in \mathbb{R}^E$ , where edge variable  $x_e^S$  takes value 1 if  $e$  appears in  $S$  and 0 otherwise. Each edge  $(v, w) \in E$  has two binary variables  $f_{v,w}^i$  and  $s_{v,w}$ :  $f_{v,w}^i$  represents the amount of flow along edge  $(v, w)$  whose destination is group  $G_i$  and variables  $s_{v,w}$  are binary selection variables denoting the presence or absence of edge  $(v, w)$  in the group Steiner tree. The ILP is:

$$\min \sum_{v,w \in V} d_{v,w} s_{v,w} \tag{1}$$

$$\text{subject to } \sum_{w \in V} f_{v,w}^i = \sum_{w \in V} f_{w,v}^i \text{ for all } v \in V \setminus (\cup_i R_i) \tag{2}$$

$$\sum_{v \in V} \sum_{t \in R_i} f_{v,t}^i = 1, \sum_{v \in V} \sum_{t \in R_i} f_{t,v}^i = 0, \sum_{v \in V} f_{r,v}^i = 1 \ \forall \text{ groups } R_i \tag{3}$$

$$0 \leq f_{v,w}^i \leq s_{v,w} \text{ for all } t \in T, \quad s_{v,w} \in \{0, 1\} \text{ for all } e \in E. \tag{4}$$

Constraints (2) impose flow conservation on all vertices not belonging to any group. Constraints (3) impose the inflow/outflow constraints on groups  $R_i$ . Finally, Constraints (4) impose the condition that there is positive flow on an edge only if the edge is selected. This ILP solves the diploid imperfect phylogeny problem to optimality.

## 6 Simulation Results

To mimic what was done in the past experimental studies [23,28], we considered only a single locus, where a locus is a collection of sites. No recombination was considered in our simulations. We implemented a forward simulator for a single locus in a diploid population of constant size  $N$  undergoing discrete-time random mating with non-overlapping generations. Given  $N$  diploid parents at generation  $t - 1$ , individuals at generation  $t$  were obtained as follows: With probability  $1 - p_s$ , one parent was randomly chosen with replacement and it produced a progeny via self-cloning. With probability  $p_s$ , a pair of parents was randomly chosen with replacement, and they produced a progeny via meiosis and mating. When producing a progeny, either by self-cloning or by sexual reproduction, new mutations were introduced according to a specified rate. This procedure was repeated until  $N$  progenies were produced for generation  $t$ .

Forward simulations are computationally intensive, so we used  $N = 1000$  to obtain simulations in a reasonable time. We started each simulation at generation 0 with  $N$  identical diploid individuals, and then ran the simulation for  $\tau = 4000$  generations with  $p_s > 0$ , followed by  $\tau_A$  generations of asexual phase (i.e., with  $p_s = 0$ ). Note that the average number of sexual reproductions in the history of the entire population is  $\tau p_s N$ . We took  $n$  diploid samples at the end of each simulation. We performed the following two different types of simulation:

- S1. Infinite-sites mutation model with the mutation rate fixed at  $5 \times 10^{-3}$  per locus. We used varying values of  $n, \tau_A$  and  $p_s$ , and performed 500 simulations for each parameter setting.
- S2. Finite-sites mutation model with homoplasy, using 25000 sites and mutation rate  $u$  per locus; the per-site mutation rate is  $u/25000$ . We fixed  $n = 40$  and used varying values of  $u, \tau_A$  and  $p_s$ . We performed 50 simulations for each parameter setting.

**Infinite-sites case (S1):** Under this ideal toy model with no recombination or homoplasy, if the input data set does not admit a DPP solution, then it would indicate that sexual reproduction has played a role in the evolutionary history. To assess our chance of detecting signatures of past sexual reproduction, we examined how often DPP solutions exist even if some amount of sexual reproduction actually took place in the evolutionary history of the population. The results are summarized in Table 1(a). These results suggest that infrequent sexual reproduction may be difficult to detect, and that the signature of past sexual reproduction may decay rather quickly with time. However, note that the chance of detecting signatures of past sexual reproduction increases with the sample size  $n$ . Likewise, the chance increases with the number of segregating sites in the sample (results not shown). Instead of looking at one or a few genes at a time, as done in the past [10,23,28], analyzing larger fractions of diploid genomes should increase the chance of detecting signatures of past sexual reproduction.

**Finite-sites case with homoplasy (S2):** To test the performance of the ILP described in Section 5.2, we analyzed data from the above-mentioned finite-sites simulation with homoplasy. We report the results obtained from solver CPLEX 12, but have also used the GNU Linear Programming Kit in order to release a free version of our software. We performed extensive testing to analyze

**Table 1.** Simulation results discussed in Section 6 (a) Proportion of data sets admitting DPP solutions in the infinite-sites simulation study S1, with the mutation rate =  $5 \times 10^{-3}$  per locus. (b) Average ratio  $q/\eta$  of the amount of homoplasy to the total number of mutating sites in the finite-sites simulation study S2, with  $n = 40$  and 25000 sites.

(a)							(b)						
$n$	$p_s$	Asexual phase $\tau_A$					$u$	$p_s$	Asexual phase $\tau_A$				
		0	100	500	1000	2000			100	500	1000	2000	
10	$1 \times 10^{-5}$	0.98	0.99	0.99	1.00	1.00	$1 \times 10^{-3}$	$1 \times 10^{-5}$	0.159	0.134	0.126	0.122	
25	$1 \times 10^{-5}$	0.96	0.99	0.99	0.99	1.00	$1 \times 10^{-3}$	$1 \times 10^{-4}$	0.212	0.181	0.096	0.091	
50	$1 \times 10^{-5}$	0.95	0.99	0.99	0.99	1.00	$1 \times 10^{-3}$	$1 \times 10^{-3}$	0.265	0.261	0.247	0.213	
75	$1 \times 10^{-5}$	0.94	0.98	0.99	0.99	1.00	$1 \times 10^{-3}$	$1 \times 10^{-2}$	0.559	0.290	0.242	0.112	
10	$1 \times 10^{-4}$	0.79	0.86	0.94	0.96	1.00	$2 \times 10^{-3}$	$1 \times 10^{-5}$	0.162	0.157	0.159	0.129	
25	$1 \times 10^{-4}$	0.67	0.79	0.93	0.96	1.00	$2 \times 10^{-3}$	$1 \times 10^{-4}$	0.179	0.169	0.132	0.124	
50	$1 \times 10^{-4}$	0.60	0.77	0.92	0.95	1.00	$2 \times 10^{-3}$	$1 \times 10^{-3}$	0.445	0.254	0.241	0.161	
75	$1 \times 10^{-4}$	0.56	0.75	0.92	0.95	1.00	$2 \times 10^{-3}$	$1 \times 10^{-2}$	0.469	0.241	0.229	0.165	
10	$1 \times 10^{-3}$	0.20	0.34	0.65	0.83	0.95	$3 \times 10^{-3}$	$1 \times 10^{-5}$	0.098	0.092	0.099	0.091	
25	$1 \times 10^{-3}$	0.08	0.20	0.57	0.80	0.95	$3 \times 10^{-3}$	$1 \times 10^{-4}$	0.115	0.105	0.119	0.069	
50	$1 \times 10^{-3}$	0.03	0.15	0.52	0.79	0.95	$3 \times 10^{-3}$	$1 \times 10^{-3}$	0.344	0.209	0.139	0.109	
75	$1 \times 10^{-3}$	0.01	0.14	0.50	0.78	0.95	$3 \times 10^{-3}$	$1 \times 10^{-2}$	0.496	0.231	0.158	0.134	
10	$1 \times 10^{-2}$	0.01	0.05	0.43	0.71	0.95	$4 \times 10^{-3}$	$1 \times 10^{-5}$	0.074	0.119	0.087	0.083	
25	$1 \times 10^{-2}$	0.00	0.01	0.29	0.65	0.94	$4 \times 10^{-3}$	$1 \times 10^{-4}$	0.147	0.136	0.109	0.098	
50	$1 \times 10^{-2}$	0.00	0.00	0.25	0.63	0.93	$4 \times 10^{-3}$	$1 \times 10^{-3}$	0.301	0.209	0.138	0.097	
75	$1 \times 10^{-2}$	0.00	0.00	0.24	0.62	0.93	$4 \times 10^{-3}$	$1 \times 10^{-2}$	0.440	0.206	0.183	0.136	

$n$  = number of diploid individuals sampled,  $p_s$  = probability of sexual reproduction,  $u$  = per-locus mutation rate.

the scaling behavior of our algorithms to larger number of sites and samples. While CPLEX is significantly faster for larger instances, GLPK is fast enough to illustrate the practicality of our algorithms on data sets of sizes of current biological interest. For each simulation instance, we used the ILP to find the minimum DIP and then calculated the imperfection  $q$  (see Definition 7) of the resulting diploid genealogy. This number corresponds to the minimum number of back or recurrent mutations needed in addition to the number of mutations  $\eta$  that would be present if the data admitted a DPP solution. As Table 1(b) shows, for each setting of the mutation rate  $u$ , increasing the probability  $p_s$  of sexual reproduction or decreasing the number of generations in the asexual phase tends to increase the mean ratio  $\frac{q}{\eta}$ . This suggests that, for a given mutation rate, the amount of detected homoplasy may provide some information about past sexual reproduction. For most of the simulations, the solver CPLEX solved the resulting ILP in fractions of a second, with the largest instance taking 1.3 seconds.

## 7 Discussion

In this paper, we considered a new problem in phylogenetics. Reconstructing the genealogy of diploid individuals is not only an interesting problem, but also has important practical applications. We believe that such a genealogical approach offers much more than can existing tests based on counting sequence differences [23, 28] or considering a single haplotype per individual [12]. To gain intuition on this new problem, we have explored algorithmic aspects of reconstructing diploid genealogies. It remains an important open problem to develop a sound statistical framework for studying the evolutionary history of asexual diploids, allowing for occasional sexual reproduction, recombination, and hybridization. Explicitly modeling the genealogy of asexual diploids will help to address a number of important questions in evolutionary biology: Could it be that sexual reproduction has actually occurred in the history of reputed ancient asexuals? If so, how big a role has sexual reproduction played in their long-term evolutionary success? If not, when was sexuality lost and how many independent times? For those species that are mainly asexual but occasionally reproduce sexually, how can we estimate the frequency of sexual reproduction? Can we distinguish the effects of mitotic recombination from that of past sexual reproduction? How does natural selection act on asexual diploids? The work described in this paper is a modest step toward addressing such questions.

As mentioned in the introduction, no significantly high level of sequence divergence was observed in the purportedly ancient asexual organism darwinulid ostrocods. It remains an open question whether this finding for ostrocods can be attributed to gene-conversion. It would be interesting to extend the work described in this paper to develop a method of reconstructing parsimonious diploid genealogies that explicitly incorporate sexual reproduction and gene-conversion. As a first step, it will be interesting to investigate whether there exists an efficient algorithm for reconstructing diploid genealogies with constrained patterns of sexual reproduction and recombination, similar to the recent work on the so-called galled-trees [15, 16, 18, 27]. Although we have focused on diploid perfect

phylogeny for two-state characters in this paper, generalizing the work to handle multi-state characters and polyploidy seems possible. (For all fixed number of states, polynomial-time algorithms exist for the haploid perfect phylogeny problem. See [1,20].)

## Acknowledgment

We thank Dan Gusfield for many helpful comments on a preliminary version of this manuscript. This research is supported in part by NIH grants K99-GM080099 (YSS) and R01-HG002942 (CHL); by NSF grants IIS-0513910 (CHL), CCF-0515378 (FL), and IIS-0803564 (FL); and by a Packard Fellowship for Science and Engineering (YSS).

## References

1. Agarwala, R., Fernández-Baca, D.: A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM J. Computing* 23, 1216–1224 (1994)
2. Bafna, V., Gusfield, D., Lancia, G., Yoosheph, S.: Haplotyping as perfect phylogeny: A direct approach. *J. Comput. Biol.* 10, 323–340 (2003)
3. Bandelt, H.J., Forster, P., Sykes, B.C., Richards, M.B.: Mitochondrial portraits of human populations using median networks. *Genetics*, 743–753 (1989)
4. Barton, N.H., Charlesworth, B.: Why sex and recombination? *Science* 281, 1986–1990 (1998)
5. Birky Jr., C.W.: Bdelloid rotifers revisited. *Proc. Nat. Acad. Sci.* 101, 2651–2652 (2004)
6. Buneman, P.: The recovery of trees from measures of dissimilarity. In: Hodson., F., et al. (eds.) *Mathematics in the Archeological and Historical Sciences*, pp. 387–395. Edinburgh University Press (1971)
7. Ding, Z., Filkov, V., Gusfield, D.: A linear-time algorithm for the perfect phylogeny haplotyping (PPH) problem. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) *RECOMB 2005*. LNCS (LNBI), vol. 3500, pp. 585–600. Springer, Heidelberg (2005)
8. Eskin, E., Halperin, E., Karp, R.: Efficient reconstruction of haplotype structure via perfect phylogeny. *J. Bioinf. Comput. Biol.* 1, 1–20 (2003)
9. Fernandez-Baca, D., Lagergren, J.: A polynomial-time algorithm for near-perfect phylogeny. *SIAM Journal on Computing* 32, 1115–1127 (2003)
10. Fontaneto, D., Herniou, E.A., Boschetti, C., Caprioli, M., Melone, G., Ricci, C., Barraclough, T.G.: Independently evolving species in asexual bdelloid rotifers. *PLoS Biology* 5(4), e87 (2007)
11. Foulds, L., Graham, R.: The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics* 3(43-49), 299 (1982)
12. Frumkin, D., Wasserstrom, A., Kaplan, S., Feige, U., Shapiro, E.: Genomic variability within an organism exposes its cell lineage tree. *PLoS Comput. Biol.* 1(5), e50 (2005)
13. Gusfield, D.: Efficient algorithms for inferring evolutionary trees. *Networks* 21, 19–28 (1991)

14. Gusfield, D.: Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In: Proc. 6th Annual Intl. Conf. on Research in Computational Molecular Biology (RECOMB), pp. 166–175 (2002)
15. Gusfield, D.: Optimal, efficient reconstruction of Root-Unknown phylogenetic networks with constrained recombination. *J. Comput. Sys. Sci.* 70, 381–398 (2005)
16. Gusfield, D., Eddhu, S., Langley, C.: Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinf. Comput. Biol.* 2, 173–213 (2004)
17. Halperin, E., Eskin, E.: Haplotype reconstruction from genotype data using Imperfect Phylogeny. *Bioinformatics* 20, 1842–1849 (2004)
18. Huynh, T.N.D., Jansson, J., Nguyen, N.B., Sung, W.-K.: Constructing a smallest refining galled phylogenetic network. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2005. LNCS (LNBI), vol. 3500, pp. 265–280. Springer, Heidelberg (2005)
19. Judson, P.O., Normark, B.B.: Ancient asexual scandals. *Trends Ecol. Evol.* 11, 41–46 (1996)
20. Kannan, S., Warnow, T.: A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed. *SIAM J. Computing* 26, 1749–1763 (1997)
21. Keightley, P.D., Otto, S.P.: Interference among deleterious mutations favours sex and recombination in finite populations. *Nature* 443, 89–92 (2006)
22. Lunt, D.: Genetic tests of ancient asexuality in root knot nematodes reveal recent hybrid origins. *BMC Evolutionary Biology* 8, 194 (2008)
23. Mark Welch, D., Meselson, M.: Evidence for the evolution of bdelloid rotifers without sexual reproduction or genetic exchange. *Science* 288, 1211–1215 (2000)
24. Martens, K., Rossetti, G., Horne, D.J.: How ancient are ancient asexuals? *Proc. R. Soc. London B* 270, 723–729 (2003)
25. Maynard Smith, J.: *The Evolution of Sex*. Cambridge University Press, Cambridge (1978)
26. Maynard Smith, J.: Contemplating life without sex. *Nature* 324, 300–301 (1986)
27. Nakhleh, L., Warnow, T., Linder, C.: Reconstructing reticulate evolution in species – theory and practice. In: Proc. 8th Annual Intl. Conf. on Research in Computational Molecular Biology (RECOMB), pp. 337–346 (2004)
28. Schön, I., Martens, K.: No slave to sex. *Proc. R. Soc. London B* 270, 827–833 (2003)
29. Semple, C., Steel, M.: *Phylogenetics*. Oxford University Press, Oxford (2003)
30. Sridhar, S., Blleloch, G.E., Ravi, R., Schwartz, R.: Optimal imperfect phylogeny reconstruction and haplotyping. In: Proceedings of Computational Systems Bioinformatics, pp. 199–210 (2006)
31. Sridhar, S., Dhamdhare, K., Blleloch, G.E., Halperin, E., Ravi, R., Schwartz, R.: Simple reconstruction of binary near-perfect phylogenetic trees. In: Proceedings of International Workshop on Bioinformatics Research and Applications, pp. 799–806 (2006)
32. Sridhar, S., Lam, F., Blleloch, G.E., Ravi, R., Schwartz, R.: Efficiently finding the most parsimonious phylogenetic tree via linear programming. In: Măndoiu, I.I., Zelikovsky, A. (eds.) ISBRA 2007. LNCS (LNBI), vol. 4463, pp. 37–48. Springer, Heidelberg (2007)
33. Vijayasatya, R., Mukherjee, A.: An efficient algorithm for perfect phylogeny haplotyping. In: Proc. IEEE Comput. Syst. Bioinform. Conf., pp. 103–110 (2005)
34. Waggoner, B.M., Poinar Jr., G.O.: Fossil habrotrichid rotifers in Dominican amber. *Experientia* 49(4), 354–357 (1993)

# Genovo: *De Novo* Assembly for Metagenomes

Jonathan Laserson, Vladimir Jojic, and Daphne Koller\*

Department of Computer Science, Stanford University, Stanford CA 94305, USA  
koller@cs.stanford.edu

**Abstract.** Next-generation sequencing technologies produce a large number of noisy reads from the DNA in a sample. Metagenomics and population sequencing aim to recover the genomic sequences of the species in the sample, which could be of high diversity. Methods geared towards single sequence reconstruction are not sensitive enough when applied in this setting. We introduce a generative probabilistic model of read generation from environmental samples and present Genovo, a novel *de novo* sequence assembler that discovers likely sequence reconstructions under the model. A Chinese restaurant process prior accounts for the unknown number of genomes in the sample. Inference is made by applying a series of hill-climbing steps iteratively until convergence. We compare the performance of Genovo to three other short read assembly programs across one synthetic dataset and eight metagenomic datasets created using the 454 platform, the largest of which has 311k reads. Genovo’s reconstructions cover more bases and recover more genes than the other methods, and yield a higher assembly score.

## 1 Introduction

Metagenomics and population sequencing aim to recover the genomic sequences in a genetically diverse environmental sample. Examples of such environments include biomes of narrow systems such as human gut [13], honey bees [8], or corals [23,19] and also larger ecosystems [24,22]. These studies advance our systemic understanding of biological processes and communities. In addition, the recovered sequences can enable the discovery of new species [24] or reveal details of poorly understood processes [26]. Another set of examples include cancer tumor cells [27] and pathogen populations such as HIV viral strains [25], where the genetic diversity is associated with disease progression and impacts the effectiveness of the drug treatment regime. Finally, the genetic structure of microbial populations may yield insight into evolutionary mechanisms such as horizontal gene transfer, and enable determination of genetic islands carrying functional toolkits necessary for survival and pathogenicity [20].

Such studies are made possible through the use of next-generation sequencing technologies, such as the Illumina Genome Analyzer (GA), Roche/454 FLX system, and AB SOLiD system. Compared to older sequencing methods, these

---

\* JL and VJ contributed equally to this work. Correspondence should be addressed to DK.



sequencers produce a much larger number of relatively short and noisy reads of the DNA in a sample, at a significantly lower cost.

While there are a few *de novo* assemblers aimed at single sequence reconstruction from short reads [6,29,15,5], there are no such tools designed specifically for metagenomics. The challenges stem from uncertainty about the population’s size and composition. Additionally, coverage across species is uneven and affected by the species’ frequency in the sample. Analysis of the complete populations requires sensitive methods that can reconstruct sequences even for the low-coverage species. Methods geared towards single sequence reconstruction are not sensitive enough when applied in this setting.

Such single sequence reconstruction tools commonly frame the problem as a search for an Eulerian path in a de Bruijn graph. The nodes of the graph are k-mers, with an edge connecting any two k-mers positioned consecutively on the same read. As mentioned by Chaisson et al. [7], “the Eulerian approach works best for error-free reads and quickly deteriorates as soon as the reads have even a small number of base-calling errors”. To cope with this problem, a large computational effort is used to detect and correct read errors before any assembly is done. While this approach is feasible for the ultra-short Illumina reads, the task becomes much harder in 454 reads, as the average read length is above 100 (and can reach 400b) and almost every read has an error. In addition, the error correction usually treats reads with low-frequency k-mers as erroneous and discards them. In metagenomics, this could filter out low-frequency species.

We introduce a generative probabilistic model of read generation from environmental samples and present Genovo, a novel *de novo* sequence assembler that works by discovering likely sequence reconstructions under the model. The model captures the uncertainty about the population structure as well as the noise model of the sequencing technology. A Chinese restaurant process prior accounts for the unknown number of genomes in the sample. To discover likely assemblies we perform a series of deterministic and stochastic hill-climbing moves, based on the iterated conditional modes (ICM) algorithm. As we show, our Bayesian approach offers a better sensitivity for assembly in highly diverse environments.

The accurate and sensitive reconstruction of populations has been tackled in restricted domains, such as HIV sequencing, both experimentally [25] and computationally [16,11,28]. However, these tools require prior information on the population and utilize a reference genome. A Chinese restaurant process, similar to ours, was also used in the recent work of Zagordi et al [28]. However, their approach is applicable only to a very small-scale ( $10^3$ ) set of reads already aligned to a short reference sequence. Our method uses no prior information, scales up to the order of  $10^5$  454 reads, and simultaneously performs read multiple alignment, read denoising and *de novo* sequence assembly.

We compare the performance of our algorithm to three state of the art short read assembly programs in terms of the number of GenBank bases covered, the number of amino acids recognized by PFAM profiles, and using a score we developed, which quantifies the quality of a *de novo* assembly using no external information. The comparison is conducted on 8 metagenomic datasets



[20,3,4,8,23,10](#) and one synthetic dataset. Genovo’s reconstructions show better performance across a variety of datasets. Genovo is publicly available online at <http://cs.stanford.edu/genovo>.

## 2 Methods

### Probabilistic Model

An assembly consists of a list of contigs, and a mapping of each read to a contiguous area in a contig. The contigs are represented each as a list of DNA letters  $\{b_{so}\}$ , where  $b_{so}$  is the letter at position  $o$  of contig  $s$ . For each read  $x_i$ , we have its contig number  $s_i$ , and its starting location  $o_i$  within the contig. We denote by  $y_i$  the alignment (orientation, insertions and deletions) required to match  $x_i$  base-for-base with the contig. Bold-face letters, such as  $\mathbf{b}$  or  $\mathbf{s}$ , represent the set of variables of that type. The subscript  $-i$  excludes the variable indexed  $i$  from the set.

Our probabilistic model can be characterized as a generative process, in which we first construct an unbounded number of contigs (each has unbounded length), then assign place holders for the beginning of reads in a coordinate system of contigs and offsets, and finally copy each read’s letters (with some noise) from the place it is mapped to in the contig. Formally, this is defined as follows:

1. Infinitely many letters in infinitely many contigs are sampled uniformly:

$$b_{so} \sim \text{Uniform}(\mathcal{B}) \quad \forall s = 1 \dots \infty, \forall o = -\infty \dots \infty$$

where  $\mathcal{B}$  is the alphabet of the bases (typically  $\mathcal{B} = \{A,C,G,T\}$ ).

2.  $N$  empty reads are randomly partitioned between these contigs:

$$\mathbf{s} \sim \text{CRP}(\alpha, N)$$

We use the Chinese Restaurant Process (CRP) [1] as a prior for the randomized partition.  $\text{CRP}(\alpha, N)$  generates a partition of  $N$  items by assigning the items to classes incrementally. If the first  $i - 1$  items are assigned to classes  $s_1 \dots s_{i-1}$ , then item  $i$  joins an existing class with a probability proportional to the number of items already assigned to that class, or it joins a new class with a probability proportional to  $\alpha$ . The likelihood of a partition under this construction is invariant to the order of the items, and thus yields the following conditional distribution:

$$p(s_i = s | \mathbf{s}_{-i}) = \frac{1}{N - 1 + \alpha} \cdot \begin{cases} N_{-i,s} & s \text{ is an existing class} \\ \alpha & s \text{ represents a new class} \end{cases}$$

Where  $N_{-i,s}$  counts the number of items, not including  $i$ , that are in class  $s$ . The parameter  $\alpha$  controls the expected number of classes, which in our case represent contigs. In the appendix we show how to set it correctly.

3. The reads are assigned a starting point  $o_i$  within each contig:

$$\begin{aligned} \rho_s &\sim \text{Beta}(1, 1 + \beta) && \forall s \text{ that is not empty} \\ o_i &\sim \mathcal{G}(\rho_s) && \forall i = 1..N \end{aligned}$$

We set  $\beta = 100$ . The distribution  $\mathcal{G}$  is a symmetric variation of geometric distribution that includes all the negative integers and is centered at 0. The parameter  $\rho_s$  controls the length of the region from which reads are generated:

$$\mathcal{G}(o; \rho) = \begin{cases} 0.5(1 - \rho)^{|o|} \rho & o \neq 0 \\ \rho & o = 0 \end{cases}$$

4. Each read is assigned a length  $l_i$ , and then its letters  $x_i$  are copied (with some mismatches) from its contig  $s_i$  starting from position  $o_i$  and according to the alignment  $y_i$  (encoding orientation, insertions and deletions):

$$\begin{aligned} l_i &\sim \mathcal{L} && \forall i = 1..N \\ x_i, y_i &\sim \mathcal{A}(l_i, s_i, o_i, \mathbf{b}, p_{ins}, p_{del}, p_{mis}) && \forall i = 1..N \end{aligned}$$

$\mathcal{L}$  is any arbitrary distribution over read lengths. The distribution  $\mathcal{A}$  represents the noise model known for the sequencing technology (454, Illumina, etc.). For example, if each read letter has a  $p_{mis}$  probability to be copied incorrectly, and the probabilities for insertions and deletions are  $p_{ins}$  and  $p_{del}$  respectively, then the log-probability  $\log p(x_i, y_i | o_i, s_i, l_i, \mathbf{b})$  of generating a read in the reverse orientation with  $n_{hit}$  matches,  $n_{mis}$  mismatches,  $n_{ins}$  insertions and  $n_{del}$  deletions is

$$\log 0.5 + n_{hit} \log(1 - p_{mis}) + n_{mis} \log\left(\frac{p_{mis}}{|\mathcal{B}| - 1}\right) + n_{ins} \log(p_{ins}) + n_{del} \log(p_{del})$$

assuming an equal chance (0.5) to appear in each orientation and an independent noise model. Given an assembly, we denote the above quantity as  $\text{score}_{\text{READ}}^i$ , where  $i$  is the read index.

This model includes an infinite number of  $b_{so}$  variables, which clearly cannot all be represented in the algorithm. The trick is to treat most of these variables as ‘unobserved’, effectively integrating them out during likelihood computations. The only observed  $b_{so}$  letters are those that are supported by reads, i.e. have at least one read letter aligned to location  $(s, o)$ . Hence, in the algorithm detailed below, if a contig letter loses its read support, it immediately becomes ‘unobserved’.

### Algorithm

Our algorithm is an instance of the iterated conditional modes (ICM) algorithm [2], which maximizes local conditional probabilities sequentially, in order to reach the MAP solution. Starting from any initial assembly (our initializing assembly

treats each read as occupying its own contig), our algorithm performs a series of hill-climbing moves in the space of assemblies, in an iterative fashion. We run our algorithm until convergence (200-300 iterations), and then we output the assembly that achieved the highest probability thus far. Running the algorithm multiple times with different random seeds showed no significant influence on the resulting assembly. This suggests that while our algorithm has some stochastic elements, the variability of the output is low. We list below the moves used to explore the space:

**Consensus Sequence.** This type of move performs ICM updates over the (observed) letter variables  $b_{so}$ . For each location  $(s, o)$ , let  $a_{so}^b$  be the number of reads in the current assembly that align the letter  $b \in \mathcal{B}$  to location  $(s, o)$ . Since we assumed a uniform prior over the contig letters, we optimize the score by setting  $b_{so} = \arg \max_{b \in \mathcal{B}} a_{so}^b$  (ties broken randomly).

**Read Mapping.** This move performs stochastic ICM updates over the read variables  $s_i, o_i, y_i$ . For each read  $i$ , we start by removing it completely from the assembly. We choose a new location and alignment for the read  $(s_i, o_i, y_i)$  by sampling from the joint posterior  $p(s_i = s, o_i = o, y_i = y | x_i, \mathbf{y}_{-i}, \mathbf{s}_{-i}, \mathbf{o}_{-i}, \mathbf{b}, \boldsymbol{\rho})$ .

For every potential location  $(s, o)$ , we first compute  $y_{so}^*$ , the best alignment of the read for that location, using the banded Smith-Waterman algorithm (applied to both read orientations):

$$y_{so}^* = \arg \max_y p(x_i, y | s_i = s, o_i = o, \mathbf{b}).$$

This includes locations where the read only partially overlaps with the contig, in which case aligning a read letter to an unobserved contig letter entails a probabilistic price of  $\log(|\mathcal{B}|^{-1})$  per letter. We now set  $s_i, o_i$  by sampling a location  $(s, o)$  from  $p(s_i = s, o_i = o, y_{so}^* | \cdot)$ :

$$p(s_i = s, o_i = o, y_{so}^* | \cdot) \propto p(s_i = s | \mathbf{s}_{-i}) p(o_i = o | s_i = s, \rho_s) p(x_i, y_{so}^* | s_i = s, o_i = o, \mathbf{b}) \\ \propto N_s \cdot \mathcal{G}(o; \rho_s) \cdot p(x_i, y_{so}^* | s_i = s, o_i = o, \mathbf{b})$$

The weights  $\{N_s\}$ , which are counting the number of reads in each sequence, encourage the read to join large contigs. As dictated by the CRP, we also include the case where  $s$  represents an empty contig, in which case we simply replace  $N_s$  with  $\alpha$  in the formula above. In that case, the  $p(x_i, y_{so}^*)$  component also simplifies to  $l_i \log(|\mathcal{B}|^{-1})$ , where  $l_i$  is the length of the read. We set  $y_i = y_{s_i o_i}^*$ .

As bad alignments render most  $(s, o)$  combinations extremely unlikely, we significantly speed up the above computation by filtering out combinations with implausible alignments. A very fast computation can detect locations that have at least one 10-mer in common with the read. This weak requirement is enough to filter out all but a few locations, making the optimization process efficient and scalable. A further speedup is achieved by caching common alignments.

**Geometric Variables.** This step performs ICM updates on the  $\rho_s$  variables. Each draw of a location  $o$  from  $\mathcal{G}(\rho_s)$  can be thought of a set of  $|o| + 1$  Bernoulli

trials with  $|o|$  failures and one success. Let  $\hat{o}_1, \dots, \hat{o}_{N_s}$  be the offsets of the reads assigned to sequence  $s$ . By a known property of the Beta distribution, it follows that  $\rho_s |\hat{o}_1, \dots, \hat{o}_{N_s} \sim \text{Beta}(1 + N_s, 1 + \beta + O_s)$  where  $O_s = \sum_{k=1}^{N_s} |\hat{o}_k|$ . We set  $\rho_s$  to  $\frac{N_s}{N_s + \beta + O_s}$ , the mode of the above distribution.

**Global Moves.** The above ICM moves are very local. To speed up convergence, we employ the following set of global moves, each one changes a set of variables at once, and hence takes a larger step in the space of assemblies. **(a) Propose indels.** If at a specific location most reads have an insertion, we propose to delete the corresponding letter in the contig and realign the reads, and accept the proposal if that improves the likelihood. For example, if out of  $n$  reads,  $a$  reads have an insertion, then after the proposed change those  $a$  reads will have one less insertion each, and  $n - a$  reads will have a new deletion. We have a similar move for deletions. **(b) Center.** We change the coordinate system of each sequence to maximize the  $p(o)$  component of the likelihood. **(c) Merge.** We merge two contigs whose ends overlap, if it improves the likelihood.

**Chimeric Reads.** Chimeric reads [17] are reads with a prefix and a suffix matching distant locations in the genome. In our algorithm, these rare corrupted reads often find their way to the edge of an assembled contig, thus interfering with the assembly process. To deal with this problem we occasionally (every 5 iterations) disassemble the reads sitting in the edge of a contig, thus allowing other correct reads or contigs to merge with it and increase the likelihood beyond that of the original state. If such a disassembled read was not chimeric, it will reassemble correctly in the next iteration, thus keeping the likelihood the same as before.

### Evaluation Metrics

Running on a set of reads, each method outputs the list of contigs that it was able to assemble from the reads. As done in previous studies [6, 18], we evaluate only contigs longer than 500bp.

Since for non-simulated data we do not have the actual list of genomes (the ‘ground truth’) that generated it, exact evaluation of *de novo* assemblies in metagenomic analysis is hard. We utilize three different indicators for the quality of an assembly. For the first indicator, we BLASTed the contigs produced by each method. Our goal was to estimate the number of genome bases that the contigs span. For each dataset, we used the BLAST hits of all the methods to compile a pool of genomes (downloaded from GenBank) that best represent the consensus among the methods. Then, for each method, each base in the pool’s genomes received a score indicating the quality of the best alignment covering it (the BLAST alignment score divided by the length of the aligned interval). We were then able to ask the question “How many pool bases were covered with a score greater than  $x$ ?”, and plot it in a graph which we call the *BLAST profile*.

The value of the reconstructed sequences lies in the information they carry about the underlying population, such as is provided by the functional annotation of the contigs. Our second indicator evaluated the assemblies based on this information. We decoded the contigs into protein sequences (in all 6 reading

frames) and annotated these sequences with PFAM profile detection tools [12]. We denote by  $\text{score}_{\text{PFAM}}$  the total number of decoded amino acids matched by PFAM profiles.

The above two indicators can be easily biased when exploring environments with sequences that are not yet in these databases, and hence our third indicator is a score that uses no external information and relies solely on the reads' consistency. Given an assembly, denote by  $S$  the number of contigs, and by  $L$  the total length of all the contigs. We measure the quality of an assembly using the expression

$$\sum_i \text{score}_{\text{READ}}^i - \log(|\mathcal{B}|)L + \log(|\mathcal{B}|)V_0S.$$

The first term penalizes for read errors and the second for contig length, embodying the trade off required for a good assembly. For example, the first term will be optimized by a naive assembly that lays each read in its own contig (without any changes), but the large number of total bases will incur a severe penalty from the second term. These two terms interact well since they represent probabilities - the first term is the (log) probability for generating each noisy read from the contig bases it aligns to, and the second term is the (log) probability for generating (uniformly) each contig letter. The third term ensures a minimal overlap of  $V_0$  bases between two consecutive reads. To see this, assume two reads have an overlap of  $V$  bases. If you split the contig into two at this position, the third term gives you a 'bonus' of  $\log(|\mathcal{B}|)V_0$ , while the second term penalizes you for  $\log(|\mathcal{B}|)V$  for adding  $V$  new bases to the assembly. Hence, we will prefer to merge the sequences iff  $V > V_0$ . We set  $V_0$  to 20.

To be able to compare the above score across different datasets, we normalized it by first subtracting from it the score of a naive assembly that puts each read in its own contig, and then dividing this difference by the total length of all the reads in the dataset. We define  $\text{score}_{\text{denovo}}$  to be this normalized score. See Appendix for another derivation of  $\text{score}_{\text{denovo}}$ , based on our model.

### 3 Results

While many sequencing technologies are gaining popularity, most of the short-read metagenomic datasets currently available have been sequenced using 454 sequencers (probably due to their longer reads), hence we focus on this technology. We compare the performance of our algorithm to three other tools: Velvet [29], EULER-SR [6] and Newbler, the 454 Life Science *de novo* assembler. Newbler was specifically designed for 454 reads and is provided with the 454 machine. Velvet and EULER-SR were designed for the shorter Illumina reads, but support 454 reads as well and are freely available.

Before testing the methods on the metagenomic datasets, we benchmarked them on a single sequence assembly task. We used run SRR024126 from NCBI short read archive, which contains 110k reads taken from *E. coli* (length 4.6Mb). Even though Genovo was not optimized for the single sequence assembly task, it performed on par with the other methods, as Table 1 shows.

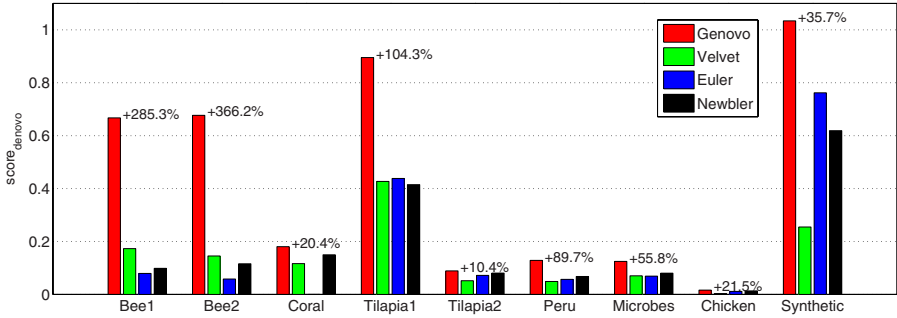
**Table 1.** Comparing the methods on a single sequencing task. Contigs were mapped using BLAST to the *E. coli* reference strand (NC\_000913.2). Coverage computed by taking the union of all matching intervals with length  $> 400$ b. Identities are exact base matches (i.e. not including gaps and mismatches).  $N_x$  is the largest value  $y$  such that at least  $x\%$  of the genome is covered by contigs of length  $\geq y$ .

	no. contigs	total contig length(kb)	N50 (kb)	N90 (kb)	coverage (%)	identities (%)
<b>Genovo</b>	129	4693	76.9	25.9	88.4	98.5
<b>Newbler</b>	150	4645	60.4	17.6	88.9	98.5
<b>Velvet</b>	621	4496	10.5	3.6	87.6	98.6
<b>Euler</b>	828	4493	7.6	2.6	86.9	98.6

**Table 2.** Metagenomic Datasets. Accession numbers starting with ‘SRR’ refer to NCBI Short Read Archive (<http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>).

name (#reads)	description (source)
Bee1(19k), Bee2(36k) <a href="#">[8]</a>	Samples from two bee colonies. Data obtained by J. DeRisi Lab.
Coral(40k) <a href="#">[23]</a>	Samples from viral fraction from whole <i>Porites compressa</i> tissue extracts (SRR001078).
Tilapia1(50k), Tilapia2(64k) <a href="#">[10]</a>	Samples from Kent SeeTech Tilapia farm containing microbial (SRR001069) and viral (SRR001066) communities isolated from the gut contents of hybrid striped bass.
Peru(84k) <a href="#">[3]</a>	Marine sediment metagenome from the Peru Margin sub-seafloor (SRR001326).
Microbes(135k) <a href="#">[4]</a>	Samples from the Rios Mesquites stromatolites in Cuatro Ciénegas, Mexico (SRR001043).
Chicken(311k) <a href="#">[20]</a>	Samples of microbiome from chicken cecum. Dataset at <a href="http://metagenomics.nmpdr.org">http://metagenomics.nmpdr.org</a> , accession 4440283.3
Synthetic(50k)	Metagenomic samples of 13 virus strains, generated using Metasim <a href="#">[21]</a> , a 454 simulator. See Appendix for list.

We carried on to compare the methods in a metagenomics setting. The comparison is conducted on 8 datasets from 6 different studies, and one synthetic dataset (see Table [2](#)). Figure 1 compares the different methods across datasets using  $\text{score}_{denovo}$  (we could not run EULER-SR on Coral). Genovo wins on every dataset, with as high as 366% advantage over the second best method. On the synthetic dataset, Genovo assembled all the reads (100.0%) into 13 contigs, one for each virus. The assemblies returned by the other methods are much more fractured — Euler, Velvet and Newbler returned 33, 47, and 38 contigs, representing only 88%, 36% and 68% of the reads, respectively. The real datasets with highest  $\text{score}_{denovo}$  were Bee1, Bee2 and Tilapia1. Genovo was able to assemble in large contigs 60%, 80% and 96% of the reads in these datasets, respectively, compared to 30%, 25% and 59% achieved by the second best method. The low  $\text{score}_{denovo}$  values for the other datasets reflect a low or no overlap between most

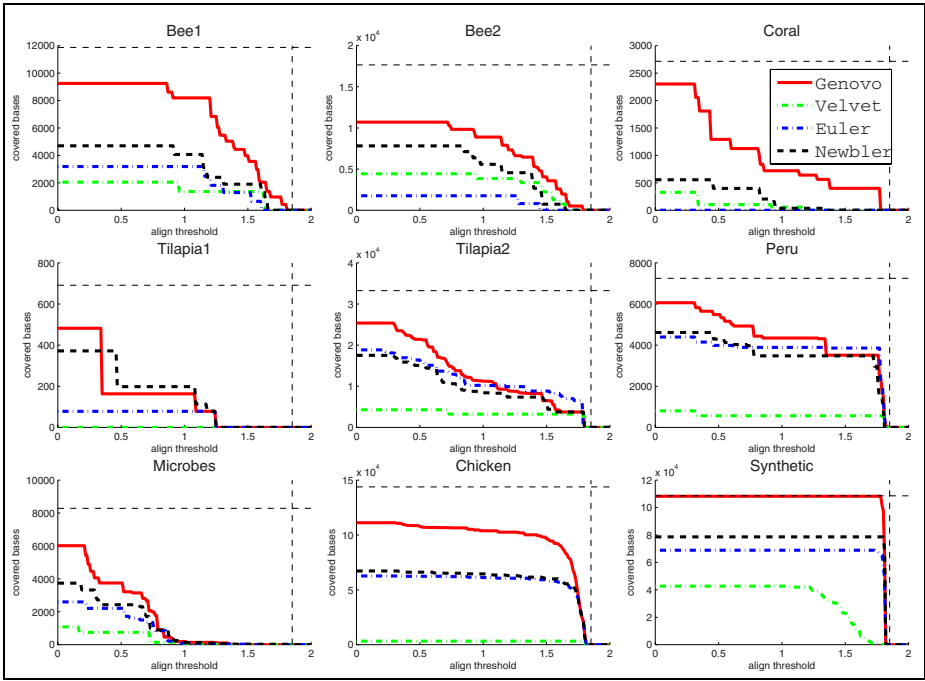


**Fig. 1.** Comparing the methods based on  $\text{score}_{\text{genovo}}$ . The numbers above the bars represent the improvement (in percentages) between Genovo and the second-best method. To compute  $\text{score}_{\text{genovo}}$ , we had to complete each list of contigs to a full assembly, by mapping each read to the location that explains it best. Reads that did not align well to any location were treated as singletons - aligned perfectly to their own contig.

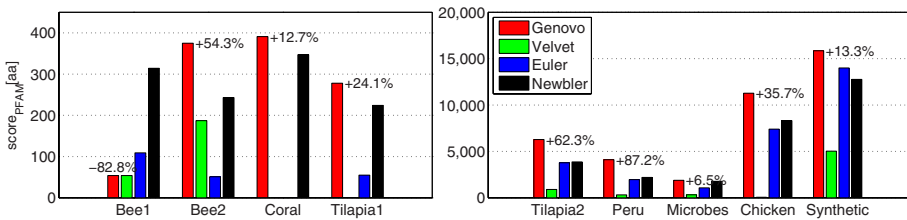
reads in those datasets. Such reads almost always lead to assemblies with many short contigs, regardless of the method used, which drive the score to 0. An example of such dataset is Chicken — all methods produced assemblies which ignored at least 97% of the reads.

Figure 2 shows the *BLAST profile* for each method, a curve that visualizes the quantity vs. the quality of the contigs (see Methods). On the synthetic dataset, Genovo covered almost all the bases (99.7%) of the 13 viruses. Other methods did poorly: Newbler, Euler and Velvet covered 72.4%, 63.4% and 39.3% of the bases, respectively. As for the real datasets, in Bee1, Bee2, Tilapia2 and Chicken many contigs showed a significant match in BLAST ( $E < 10^{-9}$ ) and the BLAST profiles provide a good indication for the assembly quality. In those cases not only does Genovo discover more bases, but it also produces better quality contigs, since Genovo’s profile dominates the other methods even on high thresholds for the alignment quality (except on Tilapia2). These differences could also translate to more species. For example, in Bee1, none of Euler’s and Newbler’s contigs matched in BLAST to *Apis mellifera* 18S ribosomal RNA gene, even though Genovo and Velvet had contigs that matched it well. On the other datasets most of the contigs did not show a significant match, and hence the genome pools compiled for those datasets are incomplete in the sense that they do not represent all the genomes in the (unknown) ground truth.

Figure 3 compares the methods in terms of the number of amino acids matched by a protein family, as measured by  $\text{score}_{\text{PFAM}}$  (see Methods). In all datasets Genovo has the highest score (with the exception of Bee1, where Newbler wins by 260aa), indicating that Genovo’s contigs hold more (and longer) annotated regions. For example, in the highly fractured Chicken dataset, our BLAST and PFAM results are markedly higher: 65% more bases were significantly ( $E < 10^{-9}$ ) matched in BLAST and 36% more amino acids recognized in PFAM compared to the second best method (Newbler). The difference is also qualitative — the



**Fig. 2.** The *BLAST* profiles of each method across all datasets. For each dataset we compiled a pool of sequences representing the ground truth. For each method, each base in the pool receives a score indicating the quality of the best alignment covering it. The curve shows how many bases received a score higher than the  $x$  value. The dashed horizontal line represents the total no. of bases in the pool covered by at least one method. The dashed vertical line represents the alignment quality of an exact match.



**Fig. 3.** Comparing the methods based on  $score_{PFAM}$ . The contigs were translated to proteins in all 6 reading frames.  $score_{PFAM}$  measures how many amino acids were recognized by protein families profilers. Due to the scale difference, results are divided into two figures with the datasets on the right figure having an order of magnitude more annotated amino acids. The numbers above the bars show the change between Genovo and the best of the other methods.



contigs reconstructed by our method were recognized by 84 distinct PFAM families, compared to 67 for Newbler’s contigs. It is important to note that in our assembly, the length of matched regions ranged from 54 to 1206aa, with average region length  $\sim 289$ aa. Similar performance on PFAM matching was achieved on the Tilapia2 dataset, where the number of matched families was 47 (compared to Newbler’s 33), and the range of matched regions was 60-1137aa. Such long matched regions could not be recovered from a read-level analysis.

The BLAST and PFAM results should not be taken as the ultimate measure of the reconstruction quality, or the dataset quality, since environmental samples may contain uncultured species that are phylogenetically distant from anything sequenced before. An example of such a dataset is Tilapia1, where almost all the contigs did not match significantly, as shown by the BLAST profiles and  $\text{score}_{\text{PFAM}}$ , even though they had significant coverage (one of our contigs, with no significant BLAST match, had a segment of 3790 bases with a minimal coverage of  $\times 85$  and a mean coverage of  $\times 177$ ). Importantly,  $\text{score}_{\text{denovo}}$  does not suffer from the same problems since it is based on the quality of the read data reconstruction, rather than the presence of a ground truth proxy.

## 4 Discussion

Metagenomic analysis involves samples of poorly understood populations. The sequenced sets of reads approximate that population and can yield information about the distribution of gene functions as well as species. However, due to fluctuations of the genomes’ coverage, these distributions may be poorly estimated. Furthermore, a read-level analysis may not be able to detect motifs that span multiple reads. Finally, a detailed analysis of events such as horizontal gene transfer will necessitate obtaining both the transposed elements and the genetic context into which they transposed. All of these concerns, in addition to a desire to obtain sequences for novel species, motivate development of sequence assembly methods aimed at problems of population sequencing.

Uncertainty over the sample composition, read coverage, and noise levels make development of methods for metagenomic sequence assembly a challenging problem. We developed a method for sequence assembly that performs well both on biologically relevant scores (based on BLAST and PFAM matches) and on a score that uses no external information. One advantage of our approach is that our probabilistic model is modular, permitting changes to the noise model without the need to modify the rest of the model. Thus, the extensions to other sequencing methodologies, as they are applied to metagenomic data, should be fairly straightforward. In addition, instead of a uniform prior over the genome letters one can use a prior based on a reference genome. Such prior will boost the model’s sensitivity in detecting variants of that genome, which can be useful when sequencing viral populations or transcriptome.

Our algorithm performs deterministic and stochastic hill-climbing moves based on the conditional probabilities derived from our probabilistic model. This

approach is suited for the problem of finding the best assembly. In a setting where the goal is to find multiple alternative reconstructions (alternative splicing, horizontal gene transfer), the same formulas can be used to construct a sampler that comprehensively explores the space according to the MCMC algorithm, and is thus more likely to explore all the modes of the distribution.

The running time required to construct an assembly can range from 15 minutes on a single CPU for a dataset with 40k reads up to a few hours for a dataset with 300k 454 reads, depending not only on the size but also on the complexity of the dataset. Newbler, Velvet and Euler typically provide their results on the order of minutes. Our increase in computational time is compatible with the time spent on a next generation sequencing run and it is worthwhile considering the superior results compared to the other assemblers.

The promise of metagenomic studies lies in their potential to elucidate interactions between members of an ecosystem and their influence on the environment they inhabit. For example, deeper understanding of constituent parts of the microbiota inhabiting humans [9,13,14] as well as their evolution in response to environmental changes, such as presence of antibiotics, will be necessary for targeted drug design. In order to begin answering questions about these populations, systematic *sequence* level analysis is necessary. With the advances of the sequencing technology and increases in the coverage, methods which can explore the space of possible reconstructions will become even more important. The model and method introduced in this paper are well suited to meet these challenges.

## Acknowledgements

This material is based upon work supported under a Stanford Graduate Fellowship and a National Science Foundation Grant BDI-0345474.

## References

1. Aldous, D.: Exchangeability and related topics. *École d'été de probabilités de Saint-Flour*, XIII, pp. 1–198 (1983)
2. Besag, J.: On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B (Methodological)* 48(3), 259–302 (1986)
3. Biddle, J.F., Fitz-Gibbon, S., Schuster, S.C., Brenchley, J.E., House, C.H.: Metagenomic signatures of the Peru Margin seafloor biosphere show a genetically distinct environment. *Proc. Natl. Acad. Sci. U.S.A.* 105, 10583–10588 (2008)
4. Breitbart, M., Hoare, A., Nitti, A., Siefert, J., Haynes, M., Dinsdale, E., Edwards, R., Souza, V., Rohwer, F., Hollander, D.: Metagenomic and stable isotopic analyses of modern freshwater microbialites in Cuatro Ciénegas, Mexico. *Environ. Microbiol.* 11, 16–34 (2009)
5. Butler, J., Mac Callum, I., Kleber, M., Shlyakhter, I.A., Belmonte, M.K., Lander, E.S., Nusbaum, C., Jaffe, D.B.: ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research* 18(5), 810–820 (2008)

6. Chaisson, M.J., Pevzner, P.A.: Short read fragment assembly of bacterial genomes. *Genome Research* 18(2), 324–330 (2008)
7. Chaisson, M.J.P., Brinza, D., Pevzner, P.A.: De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome Research* 19, 336–346 (2009)
8. Cox-Foster, D.L., Conlan, S., Holmes, E.C., Palacios, G., Evans, J.D., Moran, N.A., Quan, P.-L., Briese, T., Hornig, M., Geiser, D.M., Martinson, V., van Engelsdorp, D., Kalkstein, A.L., Drysdale, A., Hui, J., Zhai, J., Cui, L., Hutchison, S.K., Simons, J.F., Egholm, M., Pettis, J.S., Ian Lipkin, W.: A Metagenomic Survey of Microbes in Honey Bee Colony Collapse Disorder. *Science* 318(5848), 283–287 (2007)
9. Diaz-Torres, M.L., Villedieu, A., Hunt, N., McNab, R., Spratt, D.A., Allan, E., Mullany, P., Wilson, M.: Determining the antibiotic resistance potential of the indigenous oral microbiota of humans using a metagenomic approach. *FEMS Microbiol. Lett.* 258, 257–262 (2006)
10. Dinsdale, E.A., Edwards, R.A., Hall, D., Angly, F., Breitbart, M., Brulc, J.M., Furlan, M., Desnues, C., Haynes, M., Li, L., McDaniel, L., Moran, M.A., Nelson, K.E., Nilsson, C., Olson, R., Paul, J., Brito, B.R., Ruan, Y., Swan, B.K., Stevens, R., Valentine, D.L., Thurber, R.V., Wegley, L., White, B.A., Rohwer, F.: Functional metagenomic profiling of nine biomes. *Nature* 452, 629–632 (2008)
11. Eriksson, N., Pachter, L., Mitsuya, Y., Rhee, S.Y., Wang, C., Gharizadeh, B., Ronaghi, M., Shafer, R.W., Beerenwinkel, N.: Viral population estimation using pyrosequencing. *PLoS Comput. Biol.* 4, e1000074 (2008)
12. Finn, R.D., Tate, J., Mistry, J., Coghill, P.C., Sammut, S.J., Hotz, H.R., Ceric, G., Forslund, K., Eddy, S.R., Sonnhammer, E.L., Bateman, A.: The Pfam protein families database. *Nucleic Acids Res.* 36, S281–S288 (2008)
13. Gill, S.R., Pop, M., Deboy, R.T., Eckburg, P.B., Turnbaugh, P.J., Samuel, B.S., Gordon, J.I., Relman, D.A., Fraser-Liggett, C.M., Nelson, K.E.: Metagenomic analysis of the human distal gut microbiome. *Science* 312, 1355–1359 (2006)
14. Grice, E.A., Kong, H.H., Renaud, G., Young, A.C., Bouffard, G.G., Blakesley, R.W., Wolfsberg, T.G., Turner, M.L., Segre, J.A.: A diversity profile of the human skin microbiota. *Genome Res.* 18, 1043–1050 (2008)
15. Hernandez, D., Franois, P., Farinelli, L., sters, M., Schrenzel, J.: De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer. *Genome Research* 18(5), 802–809 (2008)
16. Jojic, V., Hertz, T., Jojic, N.: Population sequencing using short reads: HIV as a case study. In: *Pac. Symp. Biocomput.*, pp. 114–125 (2008)
17. Lasken, R.S., Stockwell, T.B.: Mechanism of chimera formation during the Multiple Displacement Amplification reaction. *BMC Biotechnol.* 7, 19 (2007)
18. Margulies, M., Egholm, M., Altman, W.E., Attiya, S., Bader, J.S., Bembien, L.A., Berka, J., Braverman, M.S., Chen, Y.J., Chen, Z., Dewell, S.B., Du, L., Fierro, J.M., Gomes, X.V., Godwin, B.C., He, W., Helgesen, S., Ho, C.H., Ho, C.H., Irzyk, G.P., Jando, S.C., Alenquer, M.L., Jarvie, T.P., Jirage, K.B., Kim, J.B., Knight, J.R., Lanza, J.R., Leamon, J.H., Lefkowitz, S.M., Lei, M., Li, J., Lohman, K.L., Lu, H., Makhijani, V.B., McDade, K.E., McKenna, M.P., Myers, E.W., Nickerson, E., Nobile, J.R., Plant, R., Puc, B.P., Ronan, M.T., Roth, G.T., Sarkis, G.J., Simons, J.F., Simpson, J.W., Srinivasan, M., Tartaro, K.R., Tomasz, A., Vogt, K.A., Volkmer, G.A., Wang, S.H., Wang, Y., Weiner, M.P., Yu, P., Begley, R.F., Rothberg, J.M.: Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380 (2005)

19. Meyer, E., Aglyamova, G., Wang, S., Buchanan-Carter, J., Abrego, D., Colbourne, J., Willis, B., Matz, M.: Sequencing and de novo analysis of a coral larval transcriptome using 454 gsffx. *BMC Genomics* 10(1), 219 (2009)
20. Qu, A., Brulc, J.M., Wilson, M.K., Law, B.F., Theoret, J.R., Joens, L.A., Konkel, M.E., Angly, F., Dinsdale, E.A., Edwards, R.A., Nelson, K.E., White, B.A.: Comparative metagenomics reveals host specific metavirulomes and horizontal gene transfer elements in the chicken cecum microbiome. *PLoS ONE* 3, e2945 (2008)
21. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: Metasim: A sequencing simulator for genomics and metagenomics. *PLoS ONE* 3(10), e3373 (2008)
22. Tyson, G.W., Chapman, J., Hugenholtz, P., Allen, E.E., Ram, R.J., Richardson, P.M., Solovyev, V.V., Rubin, E.M., Rokhsar, D.S., Banfield, J.F.: Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428, 37–43 (2004)
23. Vega Thurber, R.L., Barott, K.L., Hall, D., Liu, H., Rodriguez-Mueller, B., Desnues, C., Edwards, R.A., Haynes, M., Angly, F.E., Wegley, L., Rohwer, F.L.: Metagenomic analysis indicates that stressors induce production of herpes-like viruses in the coral *Porites compressa*. *Proceedings of the National Academy of Sciences* 105(47), 18413–18418 (2008)
24. Craig Venter, J., Remington, K., Heidelberg, J.F., Halpern, A.L., Rusch, D., Eisen, J.A., Wu, D., Paulsen, I., Nelson, K.E., Nelson, W., Fouts, D.E., Levy, S., Knap, A.H., Lomas, M.W., Nealson, K., White, O., Peterson, J., Hoffman, J., Parsons, R., Baden-Tillson, H., Pfannkoch, C., Rogers, Y.-H., Smith, H.O.: Environmental Genome Shotgun Sequencing of the Sargasso Sea. *Science* 304(5667), 66–74 (2004)
25. Wang, C., Mitsuya, Y., Gharizadeh, B., Ronaghi, M., Shafer, R.W.: Characterization of mutation spectra with ultra-deep pyrosequencing: application to HIV-1 drug resistance. *Genome Res.* 17, 1195–1201 (2007)
26. Warnecke, F., Luginbhl, P., Ivanova, N., Ghassemian, M., Richardson, T.H., Stege, J.T., Cayouette, M., McHardy, A.C., Djordjevic, G., Aboushadi, N., Sorek, R., Tringe, S.G., Podar, M., Martin, H.G., Kunin, V., Dalevi, D., Madejska, J., Kirton, E., Platt, D., Szeto, E., Salamov, A., Barry, K., Mikhailova, N., Kyrpides, N.C., Matson, E.G., Ottesen, E.A., Zhang, X., Hernandez, M., Murillo, C., Acosta, L.G., Rigoutsos, I., Tamayo, G., Green, B.D., Chang, C., Rubin, E.M., Mathur, E.J., Robertson, D.E., Hugenholtz, P., Leadbetter, J.R.: Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. *Nature* 450, 560–565 (2007)
27. Warren, R.L., Nelson, B.H., Holt, R.A.: Profiling model T-cell metagenomes with short reads. *Bioinformatics* 25(4), 458–464 (2009)
28. Zagordi, O., Geyrhofer, L., Roth, V., Beerenwinkel, N.: Deep sequencing of a genetically heterogeneous sample: Local haplotype reconstruction and read error correction. In: Batzoglou, S. (ed.) *RECOMB 2009*. LNCS, vol. 5541, pp. 271–284. Springer, Heidelberg (2009)
29. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18, 821–829 (2008)

## Appendix

### Understanding the Likelihood

In order to choose the  $\alpha$  parameter correctly, we have to understand our model better. Assume there are  $N$  reads and  $S$  contigs, with  $N_s$  the number of reads in contig  $s$ . Our model log-likelihood can be written as

$$\log p(\mathbf{x}, \mathbf{y}|\mathbf{s}, \mathbf{o}, \mathbf{b}) + \log p(\mathbf{b}) + \log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) + \log p(\mathbf{s})$$

where

$$\log p(\mathbf{x}, \mathbf{y}|\mathbf{s}, \mathbf{o}, \mathbf{b}) = \sum_i \text{score}_{\text{READ}}^i$$

$$\log p(\mathbf{b}) = -\log(|\mathcal{B}|)L$$

$$\log p(\mathbf{s}) = \log(\alpha)S + \sum_s \log \Gamma(N_s) + \text{const}(\alpha, N)$$

$$\log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) = \sum_s O_s \log(1 - \rho_s) + N_s \log \rho_s + \text{const}(N);$$

where  $L$  is the total length of all the contigs,  $O_s = \sum_{i:s_i=s} |o_i|$ , and  $\Gamma(\cdot)$  is the gamma function. There is an interesting interaction between  $\log p(\mathbf{s})$  and  $\log p(\mathbf{o})$ . To simplify  $\log p(\mathbf{s})$  we use the Sterling approximation  $\log \Gamma(x) \approx (x - \frac{1}{2}) \log x - x + \frac{1}{2} \log(2\pi)$ :

$$\sum_s \log \Gamma(N_s) \approx \sum_s N_s \log N_s + \frac{1}{2} \log(2\pi)S - \frac{1}{2} \sum_s \log N_s + \text{const}(N)$$

To simplify  $\log p(\mathbf{o})$ , we will assume there is a roughly uniform coverage across all contigs, with  $d$  the average distance between the  $o_i$  of two consecutive reads. It follows that contig  $s$  is roughly of length  $N_s d$ . After a centering move, the reads' offsets stretch from  $-N_s d/2$  to  $N_s d/2$ , and we can thus estimate as  $O_s = N_s^2 d/4$ . When  $\rho_s$  is updated, it is set to be

$$\rho_s = \frac{N_s}{N_s + \beta + O_s} = \frac{4}{4 + \frac{\beta}{N_s} + N_s d} \approx \frac{4}{N_s d}$$

(here we assume  $N_s \gg \beta \geq 1$ ). Using Taylor approximation:

$$\log(1 - \rho_s) \approx -\rho_s - 0.5\rho_s^2 = -\frac{4}{N_s d} - \frac{8}{N_s^2 d^2}$$

Hence:

$$\begin{aligned} \log p(\mathbf{o}|\mathbf{s}, \boldsymbol{\rho}) &= \sum_s \frac{N_s^2 d}{4} \left( -\frac{4}{N_s d} - \frac{8}{N_s^2 d^2} \right) + \sum_s N_s (\log \frac{4}{d} - \log N_s) \\ &= -\sum_s N_s \log N_s - \frac{2}{d} S + \text{const}(N, d) \end{aligned}$$

Combining the formulas for  $\log p(o)$  and  $\log p(s)$ , the most dominant term cancels out and we obtain this formula for the log-likelihood (removing constants):

$$\sum_i \text{score}_{\text{READ}}^i - \log(|\mathcal{B}|)L + \left( \log \alpha - \frac{2}{d} + \frac{1}{2} \log(2\pi) \right) S - \frac{1}{2} \sum_s \log N_s$$

As the last term is in effect very weak, this can be seen as an alternative derivation for  $\text{score}_{\text{denovo}}$ .

Consider an assembly that has two contigs with a perfect overlap of  $V_0$  bases. Now consider the assembly obtained by merging (correctly) the two overlapping contigs. For simplicity, assume both contigs have  $N_0$  reads. The difference in log-likelihood between those two assemblies  $\log p(\text{merged}) - \log p(\text{split})$  becomes zero when

$$\log \alpha = \log(|\mathcal{B}|)V_0 + \frac{1}{2} \log \left( \frac{N_0}{4\pi} \right) + \frac{2}{d}$$

We use this formula to tune  $\alpha$  appropriately. In the datasets we have,  $d$  is always larger than 2, which disables the last term. We want to merge contigs with  $N_0 = 10$  reads or more, provided that they have an overlap larger than  $V_0 = 20$  bases. Based on this formula, we set  $\alpha = 2^{40}$ , which experimentally gives better results than other values.

### Synthetic Dataset

We used Metasim with the default configuration for 454-250bp reads. The dataset was composed of the following sequences (in parenthesis, number of reads): Acidianus filamentous virus 1 (14505), Akabane virus segment L (4247), Akabane virus segment M (2636), Black queen cell virus (5309), Cactus virus X (3523), Chinese wheat mosaic virus RNA1 (3300), Chinese wheat mosaic virus RNA2 (1649), Cucurbit aphid-borne yellows virus (2183), Equine arteritis virus (4832), Goose paramyxovirus SF02 (4714), Human papillomavirus - 1 (1846), Okra mosaic virus (1016), Pariacoto virus RNA1 (240).

### Running Velvet, Euler and Newbler

For *Velvet*, we run `velveth` with k-mer length 21. We run `velvetg` multiple times using 14 values between 1 and 30 for the `-cov_cutoff` parameter. We choose the configuration which maximizes the N50. For *EULER-SR*, we run `Assemble.pl` setting the k-mer length to 25. For *Newbler*, we run `runAssembly` on the fasta file.

# MoGUL: Detecting Common Insertions and Deletions in a Population

Seunghak Lee<sup>1,2</sup>, Eric Xing<sup>2</sup>, and Michael Brudno<sup>1,3,\*</sup>

<sup>1</sup> Department of Computer Science, University of Toronto, Canada

<sup>2</sup> School of Computer Science, Carnegie Mellon University, USA

<sup>3</sup> Banting and Best Dept. of Medical Research, University of Toronto, Canada  
brudno@cs.toronto.edu

**Abstract.** While the discovery of structural variants in the human population is ongoing, most methods for this task assume that the genome is sequenced to high coverage (e.g. 40x), and use the combined power of the many sequenced reads and mate pairs to identify the variants. In contrast, the 1000 Genomes Project hopes to sequence hundreds of human genotypes, but at low coverage (4-6x), and most of the current methods are unable to discover insertion/deletion and structural variants from this data.

In order to identify indels from multiple low-coverage individuals we have developed the MoGUL (Mixture of Genotypes Variant Locator) framework, which identifies potential locations with indels by examining mate pairs generated from all sequenced individuals simultaneously, uses a Bayesian network with appropriate priors to explicitly model each individual as homozygous or heterozygous for each locus, and computes the expected Minor Allele Frequency (MAF) for all predicted variants. We have used MoGUL to identify variants in 1000 Genomes data, as well as in simulated genotypes, and show good accuracy at predicting indels, especially for  $MAF > 0.06$  and indel size  $> 20$  base pairs.

## 1 Introduction

Next generation sequencing technologies have dramatically decreased the cost of sequencing human genomes. These technologies are enabling the 1000 Genomes Project - an ambitious undertaking to reconstruct hundreds of genotypes and understand the polymorphisms present in the human population. The resequencing of humans for the 1000 Genomes Project uses a combination of approaches, including deep sequencing of several individuals and whole-exome resequencing via DNA-capture. Simultaneously, the largest fraction of individuals will be sequenced via a low-coverage whole-genome shotgun approach, where each individual will be sequenced to  $\sim 4-6x$  coverage. At this point in time it is not clear if this low coverage will be sufficient to identify a large fraction of the human variation, especially structural genomic polymorphisms.

While methods for the discovery of SNPs from read mapping have been available for some time [1], and the past two years have seen several tools developed

---

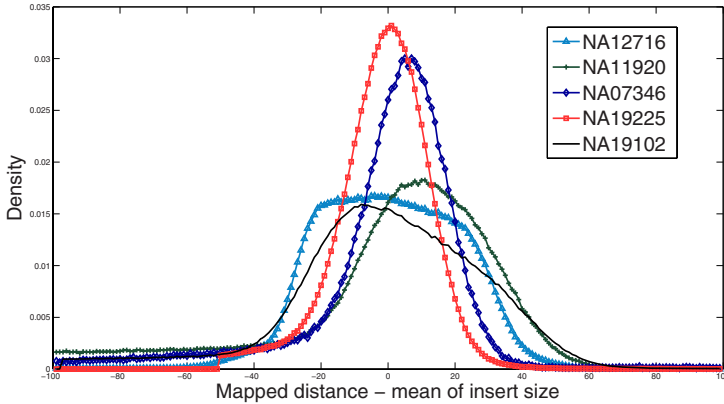
\* To whom correspondence should be addressed.

specifically for discerning SNPs from NGS data ([234](#)), the development of algorithms for the identification of larger, structural variants (SVs), including insertions and deletions (indels), is still a very active research area. While the identification of very small indels can be accomplished by directly analyzing the read mappings, with 36bp reads it is difficult to identify indels  $> 10$  bases. The identification of larger indels and other rearrangements is typically accomplished via the mate pair, or paired-end mapping technique (see [5](#) for a review). In this approach, two reads are sequenced from the two ends of a DNA fragment (the insert). Because the size of the DNA fragment is (approximately) known, structural variants can be identified by comparing the expected insert size to the distance between the mapped reads in the reference genome: if these are significantly different (the mate pair is termed discordant), it is likely that an SV has occurred between the two mappings. The past few years have seen the development of several novel methodologies and tools for SV discovery based on the analysis of discordant mate pairs, including a formal framework for identification of structural variants [6](#), tools that allow for flexible clustering of mate pairs to identify SVs [7](#), maximum parsimony and maximum likelihood approaches for SV detection [8](#), as well as tools that combine paired-end mapping with careful analysis of unpaired reads to assemble SV breakpoints [9](#).

Previously we proposed MoDIL [10](#), a method for SV identification based on the analysis of all mate pairs (concordant and discordant) that span a particular genomic location. MoDIL (Mixture of Distributions Indel Locator) fits two (possibly shifted) distributions of insert sizes (corresponding to the two haploid genotypes in a diploid) to the observed mapped distances at each location in the genome. By analyzing these distributions it is possible to discover much smaller indels than with other mate pair-based approaches. MoDIL, however, cannot be directly applied to low coverage individuals, including the bulk of the 1000 Genomes data, as it requires at least 20 inserts covering a genomic locus to identify indels (it is difficult to accurately fit two distributions with fewer data points). In the 1000 Genomes data, each locus is expected to be covered, on average, by 4 mate pairs in each individual. While the total coverage from all individuals is much higher, and most polymorphisms are di-allelic (i.e. there are only two alleles at a given locus in the human population), MoDIL expects the fractions of mate pairs sampled from each haplotype to be approximately equal. In contrast, in the 1000 Genomes data the fractions are determined by the allele frequencies and will vary across the loci.

In this work we build a Bayesian approach for the discovery of indel polymorphisms from mixtures of large numbers of genotypes, such as 1000 Genomes data. Our approach, MoGUL (Mixture of Genotypes Variant Locator), builds a Bayesian network that uses priors to explicitly model each individual as homozygous or heterozygous, and computes the expected Minor Allele Frequency (MAF) at each location along the chromosome. We use MoGUL to identify variants in the 1000 Genomes data and simulated genotypes, and demonstrate that it allows for the identification of indels  $> 30$  bases for  $MAF > 0.04$ , while indels as small as 20 bases can be identified for  $MAF > 0.06$ .





**Fig. 1.** Distribution of insert sizes from different individuals, shifted so that they are all centered at zero. Note the discrepancies among the individual distribution, necessitating modeling them as separate random variables. Here mean of insert sizes are set to be zero.

## 2 Methods

The main difficulty in identifying indels from paired-end data is differentiating mate pairs coming from a locus with an indel from those with an anomalous insert size. The insert size from each individual  $l$  follows a distribution,  $p(Y_l)$  (see Figure 1), and individual mate pairs generated from the tail of the distribution are impossible to discern from mate pairs overlapping an indel. Previous methods, such as MoDIL [10] and BreakDancer [9], use support from other mate pairs, generated by the high mate pair coverage to separate these cases. While each individual in our dataset will have only a few mate pairs sampled at every genomic location, our algorithm combines the mate pairs generated from many individuals to achieve sufficient coverage. MoGUL models mate pairs as generated from either one or two unknown distributions, corresponding to the two possible alleles at this location among the human genotypes. Our algorithm does not consider tri-allelic variants, which are rare.

Our algorithm starts by mapping all of the mate pairs onto the reference genome. We use the MrFAST tool [11], which identifies mappings for every mate pair that has at most 2 mismatches in each read and has the *mapped distance* (the distance between the forward and reverse reads of the pair) closest to the expected insert size. If this mapped distance is within 3 standard deviations, only the best mapping is identified. If no such mapping is found, all possible mappings for the two reads are returned, and our algorithm considers all of them. For every genomic location we identify those mate pairs that would be affected if that location was the site of an indel. These mate pairs will have the two reads mapping on opposite sides of the genomic location, and we refer to this set of mate pairs as a cluster (see next section).

If the genomic location is the site of an indel that is polymorphic in the human population, mate pairs in the corresponding cluster may be generated from two distributions, corresponding to the two alleles (with and without the indel). Using a Bayesian network we infer the size of the indel, as well as the individuals with indels for each cluster. Because our model may identify the same indel calls from multiple clusters, a final post-processing step is used to combine these calls and to compute the log likelihood ratio between our model and the null model. For simplicity, in the following sections we will call a mate pair “discordant” if there is significant disparity between insert size and mapped distance, and “concordant” otherwise. Note that these terms are only used for convenience – we do not *a priori* assign mate pairs to these categories.

## 2.1 Clustering Mate Pairs

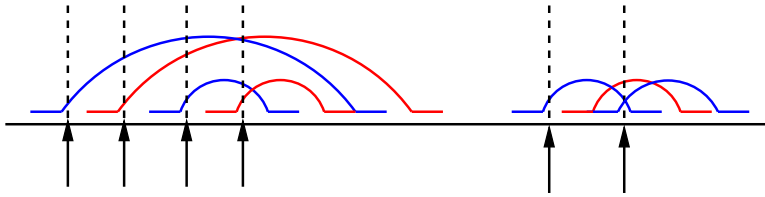
We first generate clusters with mappings of mate pairs for each genomic locus, and determine whether or not the locus contains a common indel. In this step we find a set of mate pairs  $\mathcal{C}$  from  $L$  number of individuals, all of which overlap with a particular genomic location. Figure 2 illustrates our clustering scheme.

For each mate pair we look at one base after the left read and all mate pairs overlapping the location form a cluster  $\mathcal{C}$ . We explain how we detect indels from these clusters by example. Suppose the mate pairs in Figure 2 are from the same mate pair library with the first two mate pairs discordant and the rest concordant. In such a case, as shown in Figure 2, the first two mate pairs agree on a certain indel size (they have similar mapped distance), and the indel can be detected from the second to the fourth cluster containing the two discordant mate pairs (we merge indel calls in a post-processing step).

If we use all the clusters generated by this scheme, the number of clusters will be close to the number of mate pairs, and the algorithm will be too slow. Instead, we filter out clusters if it is very likely that there is no indel at the corresponding location. For each individual  $l$ , we compute the likelihood that the mate pairs were generated from a cluster with no indel (p-value). If there is at least one individual with significant p-value ( $< 0.001$ ) or two individuals with less significant p-value ( $< 0.05$ ), the locus is deemed significant.

We define the p-value as the probability of having at least predicted size of indel ( $> \gamma$ ) given no indels. Let  $\{D_{l1}, \dots, D_{ln}\}$  represent independent and identically distributed random variables corresponding to the mapped distances of mate pairs generated from the  $l$ -th individual with insert size distribution  $p(Y_l)$ , mean  $\mu_{Y_l}$  and standard deviation  $\sigma_{Y_l}$ . Their mean follows the Gaussian distribution with mean equal to the mean of the insert size  $\mu_{Y_l}$  and standard deviation of  $\sigma_{Y_l}/\sqrt{n}$  according to the central limit theorem. We define the p-value for the individual  $l$  with the size of indel  $\gamma$  as follows:

$$\text{p-value} = \sum_{\gamma}^{\infty} P(X; 0, \sigma_{Y_l}/\sqrt{n}) = \sum_{-\infty}^0 P(X; \gamma, \sigma_{Y_l}/\sqrt{n})$$



**Fig. 2.** This figure shows how to generate clusters with mapped mate pairs in the reference genome. Mate pairs are colored by red or blue representing different individuals. For each mate pair  $X_i$ , we generate a cluster consisting of all mate pairs overlapping a genomic location of one base after the left read of the mate pair  $X_i$  (the locations of the arrows).

Here,  $X = D - \mu_{Y_i}$  is the expected size of the indel, and  $P(X)$  follows the Gaussian distribution. The second equality can be proven via symmetry of Gaussian.

In computing the p-value we correct for the possibility that the cluster contains a heterozygous indel by using a shifted sample mean:  $\gamma' = 2\gamma$ .

### 2.2 Detecting Common Indels Using a Bayesian Network

The clusters from Sec. 2.1 include mate pairs generated from many individuals, all of which have unique distributions of insert sizes (see Figure 1). We define the variable  $X_{lm}$  as the expected size of indel from the  $m$ -th mate pair of individual  $l$ :

$$X_{lm} = D_{lm} - \mu_{Y_l}$$

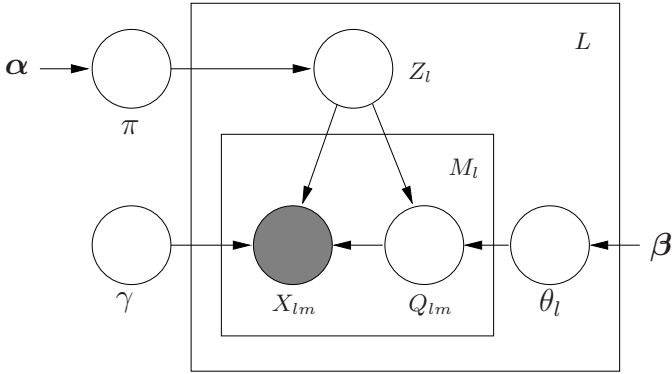
where  $D_{lm}$  is mapped distance of the  $m$ -th mate pair of the individual  $l$  and  $\mu_{Y_l}$  is mean of the insert size distribution  $p(Y_l)$ . We will use random variable  $X_{li}$  instead of  $D_{li}$  because it shifts the distributions for all individuals so that they are all centered at zero. Given a cluster of mate pairs as an input, we developed a Bayesian network (Figure 3) to infer the size of the indel polymorphism (if one exists), and haplotypes of individuals that contain the indel. The Bayesian network generates mate pairs  $\{X_{lm}\}$ , while internal states correspond to the presence/absence of indel and its heterozygosity. All random variables are defined for an input cluster, rather than the whole individual genome.

We model the individual  $l$  with random variable  $Z_l$ :

$$Z_l = \begin{cases} 0 & \text{if individual } l \text{ has no indel} \\ 1 & \text{if individual } l \text{ has an indel.} \end{cases}$$

We use the random variable  $Q_{lm}$  to model the two copies of chromosomes (alleles) in individual  $l$ . Note that subscript  $l$  refers to individual  $l$  and  $m$  denotes  $m$ -th mate pair generated from this individual:

$$Q_{lm} = \begin{cases} 0 & \text{if } Z_l = 1 \text{ and chromosome contains no indel} \\ 1 & \text{if } Z_l = 1 \text{ and chromosome contains an indel} \\ 2 & \text{if } Z_l = 0. \end{cases}$$



**Fig. 3.** Bayesian network for detecting common indels at a particular locus in the genome. Here  $L$  represents the number of individuals and  $M_l$  is the number of mate pairs from individual  $l$ . The random variable  $Z_l$  determines whether individual  $l$  has an indel or not. If individual  $l$  has an indel ( $Z_l = 1$ ),  $Q_{lm}$  generates a mate pair  $X_{lm}$  and  $\theta_l$  controls the heterozygosity of  $Z_l$ . Mate pair,  $X_{lm}$ , is generated from distribution of insert sizes with zero mean or with shifted mean of  $\gamma$  if  $X_{lm}$  has an indel. If individual  $l$  has no indel ( $Z_l = 0$ ), mate pairs  $\{X_{lm}\}_{m=1}^{M_l}$  are generated from  $p(Y_l)$  with zero mean. Priors  $\pi$  and  $\theta_l$  are controlled by  $\alpha$  and  $\beta$  parameters.

As shown in Figure 3 we can generate  $X_{lm}$  given  $Z_l$ ,  $Q_{lm}$  and size of indel  $\gamma$ . For example, if  $Q_{lm} = 1$ ,  $X_{lm}$  is generated from  $p(Y_l)$  with an indel size of  $\gamma$ . If  $\{Z_l = 0 \cup Q_{lm} = 0\}$  then  $X_{lm}$  is generated from  $p(Y_l)$  with no indel. For simplicity we omit the  $p(Y_l)$ s in Figure 3. To avoid overfitting problems we applied Bayesian priors  $\pi$  and  $\theta_l$  to  $Z_l$  and  $Q_{lm}$ , respectively.

We smooth the distribution of  $p(Y_l)$ , and define a new probability distribution of insert sizes,  $q(X_l)$ , for individual  $l$  as follows:

$$q(X_l) = \begin{cases} \sum_{k_i \sigma_{Y_l} \leq y - \mu_{Y_l} < k_{i+1} \sigma_{Y_l}} p_{Y_l}(y) & \text{if } k_i \sigma_{Y_l} \leq X_l < k_{i+1} \sigma_{Y_l} \\ \sum_{-k'_{j+1} \sigma_{Y_l} \leq y - \mu_{Y_l} < -k'_j \sigma_{Y_l}} p_{Y_l}(y) & \text{if } -k'_{j+1} \sigma_{Y_l} \leq X_l < -k'_j \sigma_{Y_l} \end{cases}$$

Here we sum  $p_{Y_l}(y)$ s over the intervals  $[k_i \sigma_{Y_l}, k_{i+1} \sigma_{Y_l})$  for deletions and  $[-k'_{j+1} \sigma_{Y_l}, -k'_j \sigma_{Y_l})$  for insertions. In our experiments, we used 10 values of  $k_i$  and  $k'_j$ s ( $i, j \in \{1, 2, \dots, 10\}$ ,  $k_1 = k'_1 = 0$ ). Probability distributions of the random variables in Figure 3 are defined as follows:

$$p(Z_l = z | \pi) = \pi^z (1 - \pi)^{1-z}$$

where  $z = 0$  if individual  $l$  has no indel and  $P(Z_l = 0) = \pi$  and  $P(Z_l = 1) = 1 - \pi$ .

$$p(Q_{lm} = q | Z_l = 1, \theta_l) = \theta_l^q (1 - \theta_l)^{1-q}$$

where  $q = 0$  if the chromosome contains no indel, and 1 otherwise. If  $Z_l = 0$ , we do not generate mate pair  $X_{lm}$  from  $Q_{lm}$  and set  $q = 2$ . We generate  $X_{lm}$  from the following distribution:

$$p(X_{lm} | Z_l, Q_{lm}, \gamma) = \begin{cases} q(X_{lm}) & \text{if } \{Z_l = 0 \cup q = 0\} \\ q(X_{lm} - \gamma) & \text{if } q = 1. \end{cases}$$

The priors  $\pi$  and  $\theta_l$  follow the beta distribution, which is the conjugate prior of binomial distributions.

To infer the states of our model, we find maximum a posteriori (MAP) solution because it is fast and deterministic. We initialize our model using heuristics (e.g.  $Q_{lm} = 1$  if  $X_{lm} > \sigma_l$ ) and random configurations, and run the model multiple times to avoid local maxima. Given current states of the model the update rules are given as follows (updated states are denoted by  $(*)$ ):

$$\pi^* = \frac{u + \alpha_1 - 1}{L + \alpha_1 + \alpha_2 - 2}$$

where  $u$  is the number of individuals with no indel. In practice we use  $\alpha = \{30, 1\}$  because most variants have a small MAF [12].

$$\theta_l^* = \frac{v + \beta_1 - 1}{M_l + \beta_1 + \beta_2 - 2}$$

where  $v$  is the number of mate pairs with  $Q_{lm} = 0$  in individual  $l$ , and we set  $\beta = \{5, 5\}$ , favoring heterozygous indels, as these are more likely under a neutral evolutionary model.

We update the random variables  $\gamma$ ,  $Z_l$  and  $Q_{lm}$  as follows:

$$\gamma^* = \arg \max_{\gamma} \prod_{l=1}^L \prod_{m=1}^{M_l} P(X_{lm} | Z_l, Q_{lm}, \gamma)$$

$$Z_l^* = \arg \max_{Z_l \in \{0,1\}} P(Z_l | \pi) \prod_{m=1}^{M_l} P(X_{lm} | Z_l, \gamma, Q_{lm}) P(Q_{lm} | Z_l, \theta_l)$$

$$Q_{lm}^* = \arg \max_{Q_{lm} \in \{0,1,2\}} P(Q_{lm} | Z_l, \theta_l) P(X_{lm} | Z_l, Q_{lm}, \gamma).$$

This algorithm is iterated, with each hidden random variable updated until the posterior probability of the model cannot be improved by the value of the threshold (e.g.  $\tau = 10^{-4}$ ).

### 2.3 Merging and Assigning Confidence to Indel Calls

In the post-processing step we merge duplicated indel calls. As shown in Sec. 2.1, a single indel may be found in multiple clusters. We merge indel calls if they meet three criteria: (1) the predicted indel regions overlap, (2) the expected size of the indel is similar ( $< \sigma_{\text{mix}}$ ), (3) the sets of individuals for whom the indel is predicted overlap. Here  $\sigma_{\text{mix}}$  is the standard deviation of insert sizes from all individuals.

To assign confidence values for every cluster we compute the log likelihood ratio  $R$  between our model and null model as follows:

$$R = \sum_{l=1}^L \sum_{m=1}^{M_l} \log P(X_{lm} | Z_l, Q_{lm}, \gamma) - \sum_{l=1}^L \sum_{m=1}^{M_l} \log P(X_{lm} | Z_l, Q_{lm}, 0).$$

We discard indel calls if the log likelihood ratio is not significantly larger than a pre-specified threshold (by default, 30).

### 3 Results

In the sections below, we use two different approaches to validate our algorithms. First, we use simulated data to evaluate how well MoGUL performs at different variant frequencies, and then use MoGUL to perform variant discovery on one chromosome of the current 1000 Genomes dataset, that includes 124 individuals sequenced at approximately 4x.

#### 3.1 Simulation Results

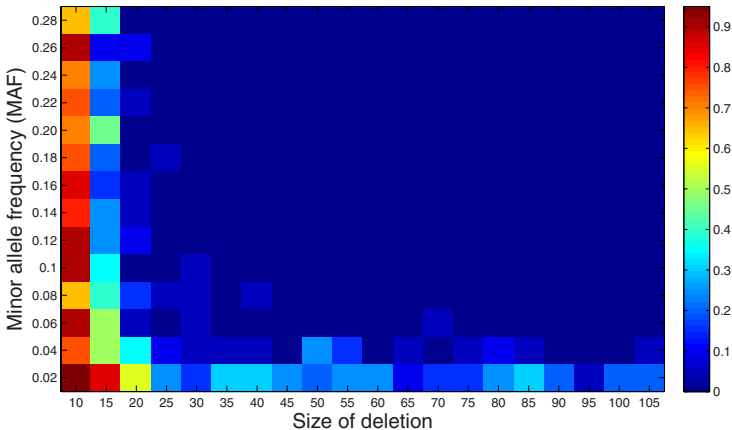
We first validate our model through simulation results. In our simulation, we sampled mate pairs from 120 individuals, with the mate pair library size of each individual  $l$  following the experimental distribution  $p(Y_l)$ .

We generated indels of 10-100 base pairs and implanted them in the individual genomes, varying the minor allele frequency (MAF) from 0.02 to 0.5. Figure 4 shows the heatmap for the performance of MoGUL. MoGUL works well for MAF greater than 0.06, for indels  $> 20$  base pairs.

To investigate the precision and recall rate of MoGUL we generated 10,000 clusters with 50 individuals (100 haplotypes). 1000 clusters contained implanted indels of 20-150 base pairs, while 100 clusters contained implanted indels of 150-1000 base pairs. For each individual we sampled mate pairs with approximately 2-3x read coverage. We detected indels for these individuals using MoGUL. The recall and precision rates of our algorithm are shown in Table 1.

#### 3.2 1000 Genome Project Pilot Dataset

In order to validate MoGUL on real data, we downloaded low coverage individuals generated by the pilot project for the 1000 Genomes project from the NCBI



**Fig. 4.** Heatmap representing the performance of MoGUL. The color of each cell indicates average error rate of 20 MoGUL simulations for a given combination of deletion size (X axis) and Minor Allele Frequencies (Y axis). If the size of indel predictions by MoGUL is more than 10bp away from the true size of deletion we consider it incorrect.

**Table 1.** Comparison between indel calls in chromosome 20 located by our approach with the datasets generated by Mills et al. [13] (all MoGUL indels), and MoDIL [10] (only indels in NA18507, the same individual as was studied by Lee et al., was considered). For the simulation experiments we consider the indel call is correct if the difference between the true indel size and the predicted one is less than 10bp and the log likelihood ratio is greater than 10.

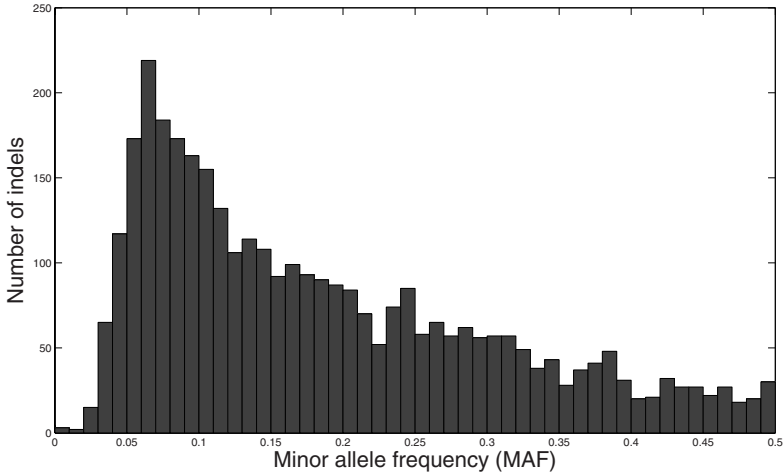
		Population			NA18507			Simulation	
Length	Type	MoGUL	Mills et al.	Overlap	MoGUL	MoDIL	Overlap	Recall	Precision
≥100bp	INS	6	20	0	2	1	1	0.91	1
	DEL	1009	183	57	34	13	10	0.89	1
50-100bp	INS	56	44	15	19	4	4	0.92	0.68
	DEL	486	71	42	22	6	5	0.86	0.99
20-50bp	INS	170	231	43	25	24	12	0.64	0.37
	DEL	1818	327	194	101	84	31	0.57	0.74

trace archive, aligned these to the NCBI reference genome with MrFAST [11], and predicted indels for all of these on chromosome 20. The results are summarized in Table 1. Overall, MoGUL predicted 3,545 events in any individual on chromosome 20. This is approximately 630 events per individual. We compare these indels to previously discovered variants both across the population, and for one specific individual, NA18507.

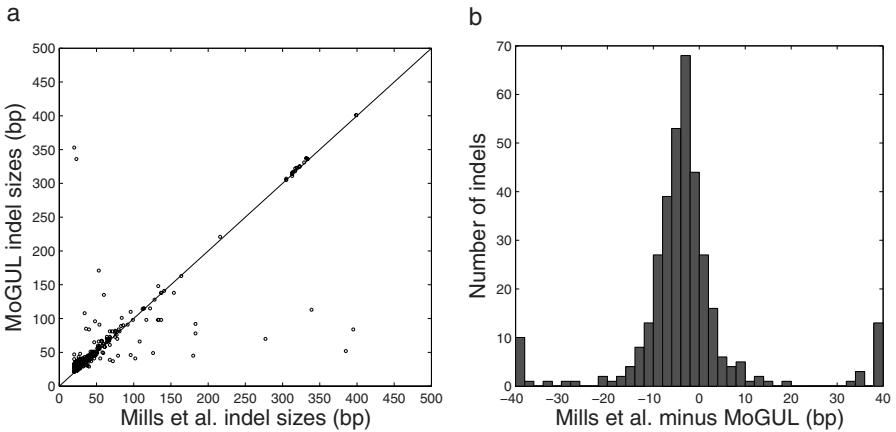
To our knowledge, the only previous study that has characterize small to medium size indels in the human populations is by Mills et al [13]. They used low coverage Sanger-style reads from 36 individuals to identify indels via the split-read mapping approach. Thus they are able to identify the exact size of the indel, while the MoGUL method infers it indirectly from the discordant mappings. Overall, the overlap between MoGUL and the indels of Mills et al was statistically significant. While exact sensitivity and specificity of the two methods is difficult to analyze, as different (and fewer) individuals were used for the Mills et al study, the size correlation of overlapping indels was very strong, and the overall error of MoGUL size estimates was small (see Figure 6).

In order to enable the direct comparison of indel discovery from single high coverage individual versus multiple low coverage individuals, we included in our dataset a down-sampled version of the NA18507 Yoruban genome which we previously analyzed using the MoDIL method. Remarkably, MoGUL was able to identify 83% of the indels > 50bp (20/24) that were previously detected by MoDIL, while identifying several additional variants that were missed by MoDIL, possibly due to low coverage in the NA18507 individual specifically. Of the events 20-50bp, 40% (43/108) were recovered by MoGUL.

In Figure 5 we plot the minor allele frequency of the variants discovered by our method. The distribution agrees with the expected curve until  $\sim$  MAF 0.07, but then drops rapidly – demonstrating MoGUL’s inability to identify indels at low minor allele frequencies.



**Fig. 5.** Distribution of minor allele frequencies for indels in the 1000 Genomes dataset



**Fig. 6.** (A) A scatter plot showing the lengths of overlapping indels between the Mills et al. dataset and MoGUL predictions. Overall the lengths are highly correlated. The cluster of indels of length 300 corresponds to Alu element activity. (B) The absolute error in the estimation of indel length. The predicted lengths of the indels are very close (typically within 10 bases) of the true indel size. Overall the distribution of the error follows a Gaussian, as expected from the model (see [10] for details). The outliers may indicate either false positives for either dataset or tri-allelic variants.

## 4 Discussion

The identification of various polymorphisms in the human population is an important step towards understanding the landscape of human genotypes. In this paper we present MoGUL: the Mixture of Genotypes Variant Locator, a tool



to identify common insertion/deletion polymorphisms from many individuals sequenced at low coverage. We validate our approach via simulated data at various allele frequencies, as well as with data from the 1000 Genomes project. MoGUL can identify indels >20 base pairs with at least 0.06 MAF, using the current low coverage data; it is expected that the coverage will double to 6–8x per individual for the final 1000 Genomes project data release, and we are hopeful that MoGUL’s performance will further improve on this larger dataset. Another application of MoGUL is resequencing of biopsy tissues, where the diseased (tumourous) tissue is biopsied (and sequenced) together with the healthy surrounding tissue, leading to a mixture of several genotypes at each location.

Simultaneously, MoGUL is only capable of recapturing a small fraction of the rare variants that predominate in the human population. While capturing common genotypes is important, it is thought that rare alleles, ones with MAF < 0.01, are much more likely to be evolutionarily harmful and disease related [12]. Designing methods that can find these variants from paired-end data, possibly incorporating direct information on read matches, as in the Pindel tool [14], is an important avenue for further research.

## Acknowledgments

We would like to thank Lisa Brooks of the NIH for permission to use 1000 genomes data, and to Can Alkan and Fereydoon Hormozdiari for providing us with the MrFAST mapping tool.

## References

1. Marth, G.T., et al.: A general approach to single-nucleotide polymorphism discovery. *Nature Genetics* 23, 452–456 (1999)
2. Li, H., Ruan, J., Durbin, R.: Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Research* 18, 1851–1858 (2008)
3. Li, R., et al.: Snp detection for massively parallel whole-genome resequencing. *Genome Research* 19, 1124–1132 (2009)
4. Hoberman, R., et al.: A probabilistic approach for SNP discovery in high-throughput human resequencing data. *Genome Research* 19, 1542–1552 (2009)
5. Medvedev, P., Stanciu, M., Brudno, M.: Computational methods for discovering structural variation with high throughput sequencing. *Nature Methods* 6, S13–S20 (2009)
6. Lee, S., Cheran, E., Brudno, M.: A robust framework for detecting structural variations in a genome. *Bioinformatics* 24, i59–i67 (2008)
7. Korbel, J., et al.: Paired-end mapping reveals extensive structural variation in the human genome. *Science* 318, 420–426 (2007)
8. Hormozdiari, F., Alkan, C., Eichler, E.E., Sahinalp, S.C.: Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Research* 19, 1270–1278 (2009)
9. Chen, K., et al.: Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature Methods* 6, 677–681 (2009)

10. Lee, S., Hormozdiari, F., Alkan, C., Brudno, M.: MoDIL: Detecting INDEL Variation with Mixtures of Distributions. *Nature Methods* 6, 473–474 (2009)
11. Alkan, C., et al.: Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature Genetics* 41, 1061–1067 (2009)
12. Kryukov, G., Shpunt, A., Stamatoyannopoulos, J., Sunyaev, S.: Power of deep, all-exon resequencing for discovery of human trait genes. *Proceedings of the National Academy of Sciences* 106, 3871–3876 (2009)
13. Mills, R., et al.: An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Research* 16, 1182–1190 (2006)
14. Ye, K., Schulz, M.H., Long, Q., Apweiler, R., Ning, Z.: Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25, 2865–2871 (2009)

# Generalized Buneman Pruning for Inferring the Most Parsimonious Multi-state Phylogeny

Navodit Misra<sup>1</sup>, Guy Blelloch<sup>2</sup>, R. Ravi<sup>3</sup>, and Russell Schwartz<sup>4</sup>

<sup>1</sup> Department of Physics, Carnegie Mellon University, Pittsburgh, USA  
`nmisra@andrew.cmu.edu`

<sup>2</sup> Computer Science Department, Carnegie Mellon University, Pittsburgh, USA  
`guyb@cs.cmu.edu`

<sup>3</sup> Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA  
`ravi@cmu.edu`

<sup>4</sup> Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, USA  
`russells@andrew.cmu.edu`

**Abstract.** Accurate reconstruction of phylogenies remains a key challenge in evolutionary biology. Most biologically plausible formulations of the problem are formally NP-hard, with no known efficient solution. The standard in practice are fast heuristic methods that are empirically known to work very well in general, but can yield results arbitrarily far from optimal. Practical exact methods, which yield exponential worst-case running times but generally much better times in practice, provide an important alternative. We report progress in this direction by introducing a provably optimal method for the weighted multi-state maximum parsimony phylogeny problem. The method is based on generalizing the notion of the Buneman graph, a construction key to efficient exact methods for binary sequences, so as to apply to sequences with arbitrary finite numbers of states with arbitrary state transition weights. We implement an integer linear programming (ILP) method for the multi-state problem using this generalized Buneman graph and demonstrate that the resulting method is able to solve data sets that are intractable by prior exact methods in run times comparable with popular heuristics. Our work provides the first method for provably optimal maximum parsimony phylogeny inference that is practical for multi-state data sets of more than a few characters.

## 1 Introduction

One of the fundamental problems in computational biology is that of inferring evolutionary relationships between a set of observed amino acid sequences or taxa. These evolutionary relationships are commonly represented by a tree (phylogeny) describing the descent of all observed taxa from a common ancestor, a reasonable model provided we are working with sequences over small enough regions or distant enough relationships that we can neglect recombination or other sources of reticulation [1]. Several criteria have been implemented in the literature for inferring phylogenies, of which one of the most popular is maximum

parsimony (MP). Maximum parsimony defines the tree(s) with the fewest mutations as the optimum, generally a reasonable assumption for short time-scales or conserved sequences. It is a simple, non-parametric criterion, as opposed to common maximum likelihood models or various popular distance-based methods [2]. Nonetheless, MP is known to be NP-hard [3] and practical implementations of MP are therefore generally based on heuristics which do not guarantee optimal solutions.

For sequences where each site or character is expressed over a set of discrete states, MP is equivalent to finding a minimum Steiner tree displaying the input taxa. For example, general DNA sequences can be expressed as strings of four nucleotide states and proteins as strings of 20 amino acid states. Recently, Sridhar *et al.* [4] used integer linear programming to efficiently find global optima for the special case of sequences with binary characters, which are important when analyzing single nucleotide polymorphism (SNP) data. The solution was made tractable in practice in large part by a pruning scheme proposed by Buneman and extended by others [5,6,7]. The so-called Buneman graph  $\mathcal{B}$  for a given set of observed strings is an induced sub-graph of the complete graph  $\mathcal{G}$  (whose nodes represent all possible strings of mutations) such that  $\mathcal{B} \subseteq \mathcal{G}$  still contains all distinct minimum Steiner trees for the observed data. By finding the Buneman graph, one can often greatly restrict the space of possible solutions to the Steiner tree problem. While there have been prior generalizations of the Buneman graph to non-binary characters [8,9], they do not provide any comparable guarantees usable for accelerating Steiner tree inference.

In this paper, we provide a new generalization of the definition of Buneman graph for any finite number of states that guarantees the resulting graph will contain all distinct minimum Steiner trees of the multi-state input set. Further, we allow transitions between different states to have independent weights. We then utilize the integer linear programming techniques developed in [4] to find provably optimal solutions to the multi-state MP phylogeny problem. We validate our method on four specific data sets chosen to exhibit different levels of difficulty: a set of nucleotide sequences from *Oryza rufipogon* [10], a set of human mt-DNA sequences representing prehistoric settlements in Australia [11], a set of HIV-1 reverse transcriptase amino acid sequences and, finally, a 500 taxa human mitochondrial DNA data set. We further compare the performance of our method, in terms of both accuracy and efficiency, with leading heuristics, PAUP\* [12] and the pars program of PHYLIP [13], showing our method to yield comparable and often far superior run times on non-trivial data sets.

## 2 Methods

### 2.1 Notation and Background

Let  $H$  be an input matrix that specifies a set of  $N$  taxa  $\chi$ , over a set of  $m$  characters  $C = \{c_1, \dots, c_m\}$  such that  $H_{ij}$  represents the  $j^{\text{th}}$  character of the  $i^{\text{th}}$  taxon. The taxa of  $H$  represent the terminal nodes of the Steiner tree inference. Further, let  $n_k$  be the number of admissible states of the  $k^{\text{th}}$  character  $c_k$ . The set

of all possible states is the space  $\mathcal{S} \equiv \{0, 1, \dots, n_1 - 1\} \otimes \dots \otimes \{0, 1, \dots, n_m - 1\}$ . We will represent the  $i^{th}$  character of any element  $b \in \mathcal{S}$ , by  $(b)_i$ . The state space  $\mathcal{S}$  can be represented as a graph  $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$  with the vertex set  $V_{\mathcal{G}} = \mathcal{S}$  and edge set  $E_{\mathcal{G}} = \{(u, v) | u, v \in \mathcal{S}, \sum_{c_p \in C}^m \delta[(u)_p, (v)_p] = 1\}$ , where  $\delta[a, b] = 0$  if  $a = b$  and 1 otherwise. Furthermore, let  $\alpha = \{\alpha_p | c_p \in C\}$  be a set of weights, such that  $\alpha_p[i, j]$  represents an edge length for a transition between states  $i, j \in \{0, \dots, n_p - 1\}$  for character  $c_p$ . We will assume that these lengths are positive (states that share zero edge length are indistinguishable), symmetric in  $i, j$  and satisfy the triangle inequality.

$$\alpha_p[i, j] + \alpha_p[j, k] \geq \alpha_p[i, k] \quad \forall \quad i, j, k \in \{0, \dots, n_p - 1\} \tag{1}$$

Non-negativity and symmetry are basic properties for any reasonable definition of length. If a particular triplet of states (say  $i, j, k$ ) does not satisfy the triangle inequality in equation 1, we can set  $\alpha_p[i, k] = \alpha_p[i, j] + \alpha_p[j, k]$  and still ensure that the shortest path connecting any set of states remains the same. We can now define a distance  $d_{\alpha}$  over  $\mathcal{G}$ , such that for any two elements  $u, v \in V_{\mathcal{G}}$

$$d_{\alpha}[u, v] \equiv \sum_{p \in C}^m \alpha_p[(u)_p, (v)_p] \tag{2}$$

Given any subgraph  $K = (V_K, E_K)$  of  $\mathcal{G}$ , we can define the length of  $K$  to be the sum of the lengths of all the edges  $L(K) \equiv \sum_{(u,v) \in E_K} d_{\alpha}[u, v]$ . The maximum parsimony phylogeny problem for  $\chi$  is equivalent to constructing the minimum Steiner tree  $T_*$  displaying the set of all specified taxa  $\chi$ , i.e., any tree  $T_*(V_*, E_*)$  such that  $\chi \subseteq V_*$  and  $L(T_*)$  is minimum. Note that  $T_*$  need not be unique.

### 2.2 Pre-processing

Before we construct the generalized Buneman graph corresponding to an input, we perform a basic pre-processing of the data. The set of taxa in the input  $H$  might not all be distinct over the length of sequence represented in  $H$ . These correspond to identical rows in  $H$  and are eliminated. Similarly, characters that do not mutate for any taxa do not affect the true phylogeny and can be removed. Furthermore, if two characters are expressed identically in  $\chi$  (modulo a relabeling of the states), we will represent them by a single character with each edge length replaced by the sum of the edge lengths of the individual characters. In case there are  $n$  such non-distinct characters, one of them is given edge lengths equal to the sum of the corresponding edges in each of the  $n$  characters and the rest are discarded. These basic pre-processing steps are often useful in considerably reducing the size of input.

### 2.3 Buneman Graph

The Buneman graph was introduced as a pruning of the complete graph for the special case of binary valued characters. For this special case it is useful

to introduce the notion of binary splits  $c_p(0)|c_p(1)$  for each character  $c_p \in C$ , which partition the set of taxa  $\chi$  into two sets  $c_p(0)$  and  $c_p(1)$  corresponding to the value expressed by  $c_p$ . Each of these sets is called a block of  $c_p$ . Each vertex of the Buneman graph  $\mathcal{B}$  can be represented by an  $m$ -tuple of blocks  $[c_1(i_1), c_2(i_2), \dots, c_m(i_m)]$ , where  $i_j = 0$  or  $1$ , for  $j \in \{1, 2, \dots, m\}$ . To construct the Buneman graph, a rule is defined for discarding/retaining the subset of vertices contained in each pair of overlapping blocks  $[c_p(i_p), c_q(i_q)]$  for each pair of characters  $(c_p, c_q) \in C \times C$ . All vertices which satisfy  $c_p(i_p) \cap c_q(i_q) = \emptyset$  for any pair of characters  $(c_p, c_q)$  can be eliminated, while those for which  $c_p(i_p) \cap c_q(i_q) \neq \emptyset$  for all  $[c_p(i_p), c_q(i_q)]$  are retained. Buneman previously established for the binary case that the retained vertex set will contain all terminal and Steiner nodes of all distinct minimum length Steiner trees.

We extend this prior result to the weighted multi-state case by presenting an algorithm analogous to the binary case to construct a graph with these properties.

### 2.4 Algorithm for Constructing the Generalized Buneman Graph

Briefly, the algorithm looks at the input matrix projected onto each distinct pair of characters  $p, q$  and constructs a  $n_p \times n_q$  matrix  $C(p, q)$ , where the  $i \times j^{th}$  element  $C(p, q)_{ij}$  is 1 only if there is at least one taxon  $t$  such that  $(t)_p = i$  and  $(t)_q = j$  and zero otherwise. The algorithm then implements a rule for each such pair of characters  $p, q$  that allows us to enumerate the possible states of those characters in any optimal Steiner tree. For clarity, we will assume that each state for each character is expressed in at least one input taxon, since states that are not present in any taxa cannot be present in a minimum length tree because of the triangle inequality. The rule is defined by a  $n_p \times n_q$  matrix  $R(p, q)$  determined by the following algorithm:

1.  $R(p, q)_{ij} \leftarrow C(p, q)_{ij}$  for all  $i \in \{0, 1, \dots, n_p - 1\}$  and  $j \in \{0, 1, \dots, n_q - 1\}$ .
2. If all non-zero entries in  $C(p, q)$  are contained in the set of elements

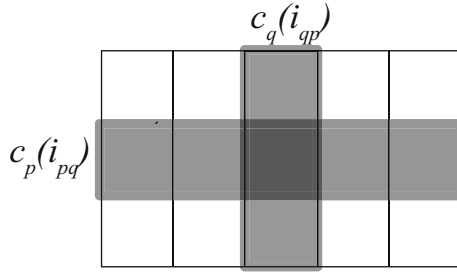
$$(\cup_k C(p, q)_{ik}) \cup (\cup_k C(p, q)_{kj})$$

for a unique pair  $i \in \{0, 1, \dots, n_p - 1\}$  and  $j \in \{0, 1, \dots, n_q - 1\}$  then  $R(p, q)_{xy} \leftarrow 1$  for all  $x, y$  such that either  $x = i$  or  $y = j$  (See Fig 1 where this pair of states are denoted  $i_{pq}$  and  $i_{qp}$ .)

3. If the condition in step 2 is not satisfied then set  $R(p, q)_{ij} \leftarrow 1$  for all  $i, j$ .

This set of rules  $\{R\}$  then defines a subgraph  $B_{pq} \subseteq \mathcal{G}$  for each pair of characters  $p, q$ , such that any vertex  $v \in B_{pq}$  if and only if  $R(p, q)_{(v)_p(v)_q} = 1$ . The intersection of these subgraphs  $\mathcal{B} = \cap_{c_p, c_q \in C} B_{pq}$  then gives the generalized Buneman graph for  $\chi$  given any set of distance metrics  $\alpha = \{\alpha_p | c_p \in C\}$ . Note that the Buneman graph of any subset of  $\chi$  is a subset of  $\mathcal{B}$ . It is easily verified that for binary characters, our algorithm yields the standard Buneman graph.

The remainder of this paper will make two contributions. First, it will show that the generalized Buneman graph  $\mathcal{B}$  defined above contains all minimum



**Fig. 1.** An example of the generalized Buneman pruning condition. If all taxa in  $\chi$  are present in the shaded region, vertices in all other blocks can be discarded.

Steiner trees for the input taxa  $\chi$ . This will in turn establish that restricting the search space for minimum Steiner trees to  $\mathcal{B}$  will not affect the correctness of the search. The paper will then empirically demonstrate the value of these methods to efficiently finding minimum Steiner trees in practice.

Before we prove that all Steiner minimum trees connecting the taxa are displayed in  $\mathcal{B}$ , we need to introduce the notion of a *neighborhood decomposition*. Suppose we are given any tree  $T(V, E)$  displaying the set of taxa  $\chi$ . We will contract each degree-two Steiner node (i.e., any node that is not present in  $\chi$ ) and replace its two incident edges by a single weighted edge. Such trees are called *X-Trees* [14]. Each X-Tree can be uniquely decomposed into its *phylogenetic X-Tree* components, which are maximal subtrees whose leaves are taxa. Formally, each phylogenetic X-Tree  $P(\psi)$  consists of a set of taxa  $\psi \subseteq \chi$  and a tree displaying them, such that there is a bijection or labeling  $\eta : l_P \rightarrow \psi$  between elements of  $\psi$  and the set of leaves  $l_P \in P(\psi)$  [14] (Fig 2). All vertices in  $P(\psi)$  with degree 3 or higher will be called *branch points*. From now on we will assume that given any input tree, such a decomposition has already been performed (Fig 2). Two phylogenetic X-Trees  $P(\psi)$  and  $P'(\psi)$  are considered *equivalent* if they have identical length and the same tree topology. By identical tree topology, we mean there is a bijection between the edge set of the two trees, such that removing any edge and its image partitions the leaves into identical bi-partitions. We define two trees to be *neighborhood distinct* if after neighborhood decomposition they differ in at least one phylogenetic X-Tree component. We define a labeling of the phylogenetic X-Tree as an injective map  $\Gamma : P \rightarrow \mathcal{G}$  between the vertices of  $P(\psi)$  and those of the graph  $\mathcal{G}$  such that  $\Gamma_u$  represents the character string for the image of vertex  $u$  in  $\mathcal{G}$ . Since leaf labels are fixed to be the character strings representing the corresponding taxa,  $\Gamma_t = \eta_t \in \psi$  for any leaf  $t \in l_P$ . Identical phylogenetic X-Trees can, however, differ in the labels  $\Gamma_u$  of internal branch points  $u \in P \setminus l_P$ .

We will use a generalization of the Fitch-Hartigan algorithm to weighted parsimony proposed by Erdos and Szekely [15,16]. The algorithm uses a similar forward pass/backward pass technique to compute an optimal labeling for any phylogenetic X-Tree  $T(\psi)$ . Arbitrarily root the tree  $T(\psi)$  at some taxon  $\zeta$  and



**Fig. 2.** An input tree and its phylogenetic X-Tree components, with taxa labelled by integers

starting with the leaves compute the minimum length  $minL(\Gamma_b, T_b)$  of any labeling of the subtree  $T_b$  consisting of the vertex  $b$  and its descendants, where the root  $b$  is labeled  $\Gamma_b$  as follows.

1. If  $\Gamma_b$  labels a leaf  $\eta_b \in \psi$ ,  $minL(\Gamma_b = \eta_b, T_b) = 0$  and  $\infty$  otherwise.
2. If  $b$  has  $k$  children  $D_b = \{v_1, \dots, v_k\}$ , and  $T_v$  is the subtree consisting of  $v \in D_b$  and its descendants,

$$minL(\Gamma_b, T_b) = \sum_{v \in D_b} \min_{\Gamma_v} \{minL(\Gamma_v, T_v) + d_\alpha[\Gamma_b, \Gamma_v]\} \tag{3}$$

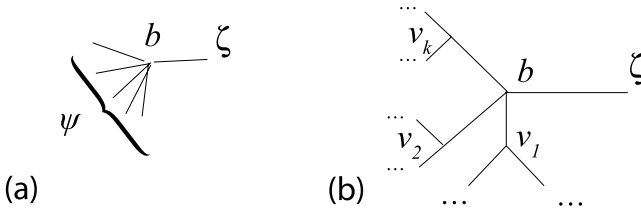
where the minimum is to be taken over all possible labels  $\Gamma_v$  for each character and for each child  $v \in D_b$ .

The optimal labeling of  $T(\psi)$  is one which minimizes the length at the root:  $L(T) = minL(\eta_\zeta, T_\zeta)$ . Labels for each descendant are inferred in a backward pass from the root to the leaves and using equation 3. Note that the minimum length of a tree is just the sum of minimum lengths for each character, i.e.,  $minL(\Gamma_b, T_b) = \sum_{c_s \in C} minL(\Gamma_b, T_b)^{(s)}$ , where  $minL(\Gamma_b, T_b)^{(s)}$  is the minimum cost of tree  $T_b$  rooted at  $b$  for character  $c_s$ .

Briefly, our proof is structured as follows: Given any phylogenetic X-Tree  $T(\psi)$  labeling (typically denoted  $\Gamma$  below), we will show that the generalized Buneman pruning algorithm for each pair of characters  $(c_p, c_q)$  defines a subgraph  $B_{pq}$  which contains at least one possible labeling of no higher cost (typically denoted  $\Phi$  below) for  $T(\psi)$ . We will then show that the intersection of these subgraphs  $\mathcal{B} = \cap_{p \neq q} B_{pq}$  thus contains an optimal labeling for  $T(\psi)$ .

If the pruning condition in step 2 of the algorithm that defines the Buneman graph is not implemented for the pair of characters  $(c_p, c_q)$ , then  $B_{pq} = \mathcal{G}$  and all labels are necessarily inside  $B_{pq}$ . We prove the following lemma for the case when the pruning condition is satisfied, i.e., there exist unique states  $i_{pq}$  of  $c_p$  and  $i_{qp}$  of  $c_q$ , such that each element in the set of leaves  $l_T = \{t \in T(\psi) | \eta_t \in \psi\}$  either has  $(\eta_t)_p = i_{pq}$  or  $(\eta_t)_q = i_{qp}$  or both. Each time we relabel vertices, we will keep all characters except  $c_p$  and  $c_q$  fixed. To economize our notation, we will represent the sum of costs in  $c_p$  and  $c_q$  of the tree  $T$  labeled by  $\Gamma$ , which has some branch point  $b$  as the root, simply by writing  $L(\Gamma, T) = L(\Gamma, T)^{(p)} + L(\Gamma, T)^{(q)}$ . We use the notation  $\Gamma_x = [(\Gamma_x)_p, (\Gamma_x)_q]$  to represent the label for a vertex  $x$  and suppress the state of all other characters.





**Fig. 3.** (a) The base case of a degree  $|\psi|$  star that can be attached to a parent vertex  $\zeta$  in the Erdos-Szekely algorithm. (b)  $T(\psi)$  for the general case (see Lemma 1).

**Lemma 1.** *Given any phylogenetic X-Tree  $T(\psi)$  with  $\psi \subseteq B_{pq}$ , and a labeling  $\Gamma$ , such that an internal branch point  $b \in T \setminus l_T$  is labeled outside  $B_{pq}$ , i.e.,  $\Gamma_b \notin B_{pq}$ , there exists an alternate labeling  $\Phi$  of  $T(\psi)$  inside  $B_{pq}$  such that*

1. either  $L(\Gamma, T) \geq L(\Phi, T) + d_\alpha[\Gamma_b, \Phi_b]$ , or —
2.  $L(\Gamma, T) \geq L(\Phi, T)$  for each of the following choices:  $\Phi_b = [i_{pq}, i_{qp}]$  or  $[i_{pq}, (\Gamma_b)_q]$  or  $[(\Gamma_b)_p, i_{qp}]$ , and  $\Phi_v = \Gamma_v$  for all  $v \neq b$ . We will call a tree that satisfies this second condition a  $(c_p, c_q)$ -Tree

*Proof.* We will use induction on the number of internal branch points outside  $B_{pq}$  to prove the claim. Without loss of generality we can consider all branch points of  $T(\psi)$  to be labeled outside  $B_{pq}$ . If some branch points are labeled inside  $B_{pq}$  then they can be treated as leaves of smaller X-Tree(s) that have all branch points outside  $B_{pq}$ . This is similar to the neighborhood decomposition we performed earlier for those branch points that were present in the set of input taxa. The set of branch points is then the set  $T \setminus l_T = \{u \in T | \Gamma_u \notin B_{pq}\}$ .

For the base case assume all the leaves are joined at a single branch point  $b$  to form a star of degree  $|\psi|$  (see Fig. 3(a) without the root  $\zeta$ ). We can group the leaves into three sets:

1.  $I = \{\eta_u = [i_{pq}, y_u] | y_u \neq i_{qp}, \eta_u \in \psi\}$
2.  $II = \{\eta_v = [x_v, i_{qp}] | x_v \neq i_{pq}, \eta_v \in \psi\}$
3.  $III = \{\eta_w = [i_{pq}, i_{qp}] | \eta_w \in \psi\}$

The cost of the tree for  $c_p$  and  $c_q$ , with branch point  $\Gamma_b = [x, y]$ , is

$$\begin{aligned}
 L(\Gamma, T)^{(p)} + L(\Gamma, T)^{(q)} &= \sum_{u \in I} (\alpha_p[x, i_{pq}] + \alpha_q[y, y_u]) + \sum_{v \in II} (\alpha_p[x, x_v] \\
 &\quad + \alpha_q[y, i_{qp}]) + \sum_{w \in III} (\alpha_p[x, i_{pq}] + \alpha_q[y, i_{qp}]) \quad (4)
 \end{aligned}$$

The only way for  $L(\Gamma, T)^{(p)} + L(\Gamma_b, T)^{(q)}$  to be minimum with  $x \neq i_{pq}$  and  $y \neq i_{qp}$ , is if  $III = \emptyset$  and  $|I| = |II|$ . For contradiction, suppose  $|I| + |III| > |II|$ . We could then define a labeling  $\Phi$  identical to  $\Gamma$  over all characters, except  $\Phi_b = [i_{pq}, y]$ , such that  $d_\alpha[\Gamma_b, \Phi_b] = \alpha_p[\Gamma_b, \Phi_b]$ . We could then reduce the length, since

$$\begin{aligned}
 L(\Gamma, T)^{(p)} &= \sum_{u \in I} \alpha_p[x, i_{pq}] + \sum_{v \in II} \alpha_p[x, x_v] + \sum_{w \in III} \alpha_p[x, i_{pq}] \\
 &\geq \alpha_p[x, i_{pq}] + \sum_{v \in II} (\alpha_p[x, x_v] + \alpha_p[x, i_{pq}]) \\
 &\geq \alpha_p[x, i_{pq}] + \sum_{v \in II} \alpha_p[i_{pq}, x_v] = L(\Phi, T)^{(p)} + d_\alpha[\Gamma_b, \Phi_b] \quad (5)
 \end{aligned}$$

where the last inequality follows from the triangle inequality. Similarly, if  $|II| + |III| > |I|$ , we could define  $\Phi_b = [x, i_{qp}]$  and arrive at  $L(\Gamma, T)^{(q)} \geq L(\Phi, T)^{(q)} + d_\alpha[\Gamma_b, \Phi_b]$ .

On the other hand if  $|I| = |II|$  and  $III = \emptyset$  setting  $\Phi_b = [i_{pq}, y]$  or  $\Phi_b = [x, i_{qp}]$  or  $\Phi_b = [i_{pq}, i_{qp}]$  all achieve a length no more than  $L(\Gamma, T)^{(p)} + L(\Gamma, T)^{(q)}$ . Therefore, this is a  $(c_p, c_q)$ -Tree. This proves the base case for our proposition.

We will now assume that the claim is true for all trees with  $n$  branch points or less. Suppose we have a labeled tree  $T(\psi)$  with  $n + 1$  branch points which are all outside  $B_{pq}$ . Let  $D_b = \{v_1, \dots, v_k\}$  be the children of a branch point  $b$  in  $T(\psi)$  and  $\{T_1, \dots, T_k\}$  be the subtrees of each  $v \in D_b$  and their descendants. Note that some of these descendants may be leaves. Since  $T(\psi)$  has at least two branch points, one of its descendants (say  $v_1$ ) must be a branch point (Fig 3(b)). Let  $T_b = T \setminus T_1$  be the subtree consisting of  $b$  and all its other descendants. For clarity we will use the notation  $\Gamma_b = [x_b, y_b]$  and  $\Gamma_{v_1} = [x_1, y_1]$ . This implies,

$$\begin{aligned}
 L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + d_\alpha[\Gamma_b, \Gamma_{v_1}] \\
 &= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \quad (6)
 \end{aligned}$$

There are four possibilities.

1. Both  $T_b$  and  $T_1$  are  $(c_p, c_q)$ -Trees with  $n$  or less branch points - In this case, by induction, both  $T_b$  and  $T_1$  can be relabeled with  $\Phi_b$  and  $\Phi_{v_1}$  of the form  $[i_{pq}, i_{qp}]$ . Since the cost in  $c_p$  and  $c_q$  of the edge  $(b, v_1)$  is now zero, we have an optimal labeling of  $T(\psi)$  within  $B_{pq}$  and  $L(\Gamma, T) \geq L(\Phi, T)$ . Note that each of the choices of the form  $[i_{pq}, y_1]$  or  $[x_1, i_{pq}]$  for relabeling of  $b$  also satisfy property 2 of the claim. Therefore, this is a  $(c_p, c_q)$ -Tree.
2.  $T_b$  is a  $(c_p, c_q)$ -Tree, but  $T_1$  is not. Therefore, there is a labeling  $\Phi$  of  $T_1$  with either  $\Phi_{v_1} = [i_{pq}, y_1]$  and/or  $\Phi_{v_1} = [x_1, i_{pq}]$  such that

$$L(\Gamma, T_1) \geq L(\Phi, T_1) + d_\alpha[\Gamma_{v_1}, \Phi_{v_1}] \quad (7)$$

Let us assume for concreteness that  $\Phi_{v_1} = [i_{pq}, y_1]$ . It will become clear that the argument works for the other possible choices. Since,  $T_b$  is a  $(c_p, c_q)$ -Tree, by induction, we can choose a labeling of  $T_b$  with  $\Phi_b = [i_{pq}, y_b]$ , such that  $L(\Gamma, T_b) \geq L(\Phi, T_b)$ . This gives

$$\begin{aligned} L(\Phi, T) &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Phi_b, \Phi_{v_1}] \\ &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{8}$$

Comparing the previous two equations with equation [6](#), we get,

$$\begin{aligned} L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_{v_1}, \Phi_{v_1}] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_p[x_1, i_{pq}] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\Phi, T_b) + L(\Phi, T_1) + \alpha_p[x_b, i_{pq}] + \alpha_q[y_b, y_1] \\ &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + d_\alpha[\Phi_b, \Phi_{v_1}] \\ &= L(\Phi, T) + d_\alpha[\Gamma_b, \Phi_b] \end{aligned} \tag{9}$$

which satisfies the first possibility of the claim. It should be clear that if  $\Phi_{v_1} = [x_1, i_{qp}]$  then the choice  $\Phi_b = [x_b, i_{qp}]$  would give an identical bound.

3.  $T_1$  is a  $(c_p, c_q)$ -Tree, but  $T_b$  is not. This case is similar to the previous one. Since  $T_b$  has less than  $n$  branch points, which are all outside  $B_{pq}$ , and it is not a  $(c_p, c_q)$ -Tree, we have from induction a labeling  $\Phi$  of  $T_b$  with either  $\Phi_b = [i_{pq}, y_b]$  and/or  $\Phi_b = [x_b, i_{pq}]$  such that

$$L(\Gamma, T_b) \geq L(\Phi, T_b) + d_\alpha[\Gamma_b, \Phi_b] \tag{10}$$

As before, let us assume  $\Phi_b = [i_{pq}, y_b]$  for concreteness. Since  $T_1$  is a  $(c_p, c_q)$ -Tree, we can choose a labeling with  $\Phi_{v_1} = [i_{pq}, y_1]$  such that  $L(\Gamma, T_1) \geq L(\Phi, T_1)$ . This gives,

$$\begin{aligned} L(\Phi, T) &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Phi_b, \Phi_{v_1}] \\ &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{11}$$

Comparing the previous two equations with equation [6](#), we get,

$$\begin{aligned} L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\ &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + d_\alpha[\Phi_b, \Phi_{v_1}] \\ &= L(\Phi, T) + d_\alpha[\Gamma_b, \Phi_b] \end{aligned} \tag{12}$$

An identical argument carries through if  $\Phi_b = [x_b, i_{qp}]$ .

4. Neither  $T_1$  or  $T_b$  are  $(c_p, c_q)$ -Trees. It follows from induction that there is a labeling  $\Phi$  such that  $L(\Gamma, T_b) \geq L(\Phi, T_b) + d_\alpha[\Gamma_b, \Phi_b]$  and  $L(\Gamma, T_1) \geq L(\Phi, T_1) + d_\alpha[\Gamma_{v_1}, \Phi_{v_1}]$ . There are two possibilities in this case.

- (a) ( $\Phi_b = [i_{pq}, y_b]$  and  $\Phi_{v_1} = [i_{pq}, y_1]$ ) or ( $\Phi_b = [x_b, i_{qp}]$  and  $\Phi_{v_1} = [x_1, i_{qp}]$ ). As before, we will prove the claim for the former possibility while the later case can be proved by an identical argument.

$$\begin{aligned} L(\Phi, T) &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Phi_b, \Phi_{v_1}] \\ &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_q[y_b, y_1] \end{aligned} \tag{13}$$

$$\begin{aligned}
 L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\
 &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + d_\alpha[\Gamma_{v_1}, \Phi_{v_1}] \\
 &\quad + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\
 &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + \alpha_q[y_b, y_1] \\
 &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + d_\alpha[\Phi_b, \Phi_{v_1}] \\
 &= L(\Phi, T) + d_\alpha[\Gamma_b, \Phi_b]
 \end{aligned} \tag{14}$$

This also satisfies the claim. The proof for  $\Phi_b = [x_b, i_{qp}]$  and  $\Phi_{v_1} = [x_1, i_{qp}]$  is identical.

- (b) ( $\Phi_b = [i_{pq}, y_b]$  and  $\Phi_{v_1} = [x_1, i_{qp}]$ ) or ( $\Phi_b = [x_b, i_{qp}]$  and  $\Phi_{v_1} = [i_{pq}, y_1]$ ). As before, we show the calculation for the former possibility. In this case

$$\begin{aligned}
 L(\Phi, T) &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Phi_b, \Phi_{v_1}] \\
 &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_p[x_b, i_{pq}] + \alpha_q[i_{qp}, y_1]
 \end{aligned} \tag{15}$$

Combining this with equation 6 we get,

$$\begin{aligned}
 L(\Gamma, T) &= L(\Gamma, T_b) + L(\Gamma, T_1) + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\
 &\geq L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Gamma_b, \Phi_b] + d_\alpha[\Gamma_{v_1}, \Phi_{v_1}] \\
 &\quad + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\
 &= L(\Phi, T_b) + L(\Phi, T_1) + \alpha_p[x_b, i_{pq}] + \alpha_q[i_{qp}, y_1] \\
 &\quad + \alpha_p[x_b, x_1] + \alpha_q[y_b, y_1] \\
 &\geq L(\Phi, T_b) + L(\Phi, T_1) + \alpha_p[x_b, i_{pq}] + \alpha_q[i_{qp}, y_1] \\
 &= L(\Phi, T_b) + L(\Phi, T_1) + d_\alpha[\Phi_b, \Phi_{v_1}] = L(\Phi_b, T)
 \end{aligned} \tag{16}$$

But if we now relabel  $b$  and  $v_1$  with  $\tilde{\Phi}_{v_1} = [i_{pq}, i_{qp}]$  and  $\tilde{\Phi}_b = [i_{pq}, i_{qp}]$  while  $\tilde{\Phi}_v = \Phi_v$  for all other  $v$ , we get  $L(\Phi, T_1) + \alpha_q[y_1, i_{qp}] \geq L(\tilde{\Phi}_{v_1}, T_1)$  and  $L(\Phi, T_b) + \alpha_p[x_b, i_{pq}] \geq L(\tilde{\Phi}, T_b)$ . This immediately gives,

$$\begin{aligned}
 L(\tilde{\Phi}, T) &= L(\tilde{\Phi}, T_b) + L(\tilde{\Phi}, T_1) + d_\alpha[\tilde{\Phi}_b, \tilde{\Phi}_{v_1}] \\
 &\geq L(\Phi, T) \geq L(\Gamma, T)
 \end{aligned} \tag{17}$$

Identical arguments work for the choices  $\tilde{\Phi}_{v_1} = [x_1, i_{qp}]$  and  $\tilde{\Phi}_b = [x_b, i_{qp}]$ .

This proves that if either of the two possibilities claimed are always true for an X-Tree with  $n$  branch points or less then they are also true for a tree with  $n + 1$  branch points. The proof for arbitrary  $n$  follows from induction. □

**Corollary 1.** *Given a minimum length phylogenetic X-Tree  $T(\psi)$  there is an optimal labeling for each branch point within  $\mathcal{B}$ .*

*Proof.* Lemma 1 establishes that for any minimum Steiner tree labeled by  $\Gamma$  and any branch point  $b \in T$  such that  $\Gamma_b \notin B_{pq}$ , an alternative optimal labeling  $\Phi$  exists such that  $\Phi_b$  is inside the union of blocks

$$\Lambda(\Gamma_b, p, q) \equiv [c_p(i_{pq})c_q(i_{qp})] \cup [c_p(i_{pq})c_q((\Gamma_b)_q)] \cup [c_p((\Gamma_b)_p)c_q(i_{qp})]$$

If we root the tree at  $b$ , the new optimal labeling for all its descendants is inferred in a backward pass of the Erdos-Szekely algorithm. This ensures that each branch point in a minimum length phylogenetic X-Tree is labeled inside  $B_{pq}$ . Let  $S_b = \cap_{B_{pq} \neq \mathcal{G}} \Lambda(\Gamma_b, p, q) \subseteq \mathcal{B}$ , where the intersection is taken over all pair of characters for which the pruning condition is satisfied. It follows from Lemma 1 that  $S_b$  also contains an alternate optimal labeling of  $T(\psi)$ . Note that  $S_b$  is a non-empty subset of  $\mathcal{B}$ . This must be true because given a character pair  $c_p, c_q$ , each union of blocks contains at least one taxon and so the rule matrix  $R(p, q)$  that defines the Buneman graph must have ones for each of these blocks. Therefore each element in  $S_b$  represents a distinct vertex of the Buneman graph.  $\square$

As argued before, any minimum Steiner tree can be decomposed uniquely into phylogenetic X-Tree components and the previous corollary ensures that each phylogenetic X-Tree can be labeled optimally inside the generalized Buneman graph. It follows that all distinct minimum Steiner trees are contained inside the generalized Buneman graph.

### 2.5 Integer Linear Program (ILP) Construction

We briefly summarize the ILP flow construction used to find the optimal phylogeny. We convert the generalized Buneman graph into a directed graph by replacing an edge between vertices  $u$  and  $v$  with two directed edges  $(u, v), (v, u)$  each with weight  $w_{uv}$  as determined by the distance metric. Each directed edge has a corresponding binary variable  $s_{u,v}$  in our ILP. We arbitrarily choose one of the taxa as the root  $r$ , which acts as a source for the flow model. The remaining taxa  $T \equiv \chi - \{r\}$  correspond to sinks. Next, we set up real-valued flow variables  $f_{u,v}^t$ , representing the flow along the edge  $(u, v)$  that is intended for terminal  $t$ . The root  $r$  outputs  $|T|$  units of flow, one for each terminal. The Steiner tree is the minimum-cost tree satisfying the flow constraints. This ILP was described in [4], and we refer the reader to that paper for further details. The ILP for this construction of the Steiner tree problem is the following:

$$\begin{aligned} & \text{Minimize } \sum_{(u,v) \in \mathcal{B}} w_{uv} s_{u,v} \\ & \text{subject to } \sum_v (f_{u,v}^t - f_{v,u}^t) = 0 \quad \forall u \in \mathcal{B} \setminus \{t, r\}, \forall t \in T \\ & \quad \sum_v (f_{r,v}^t - f_{v,r}^t) = 1 \quad \forall t \in T \\ & \quad 0 \leq f_{u,v}^t \leq s_{u,v} \quad \forall (u, v) \in \mathcal{B}, \forall t \in T \\ & \quad s_{u,v} \in \{0, 1\} \quad \forall (u, v) \in \mathcal{B} \end{aligned} \tag{18}$$

**Table 1.** Pruning and run time results for the data sets reported

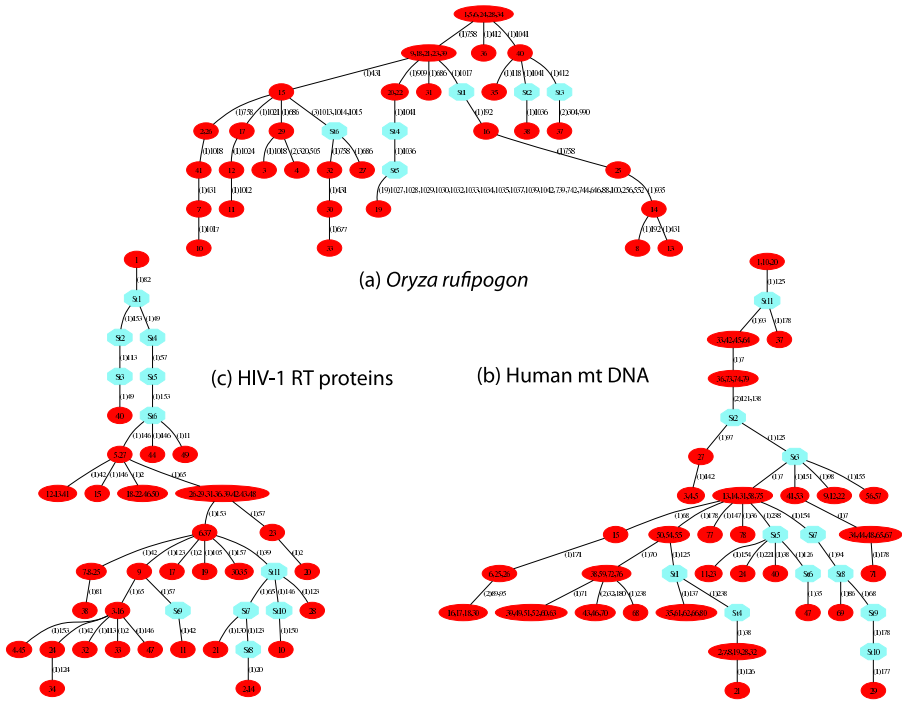
Data	Input (raw)	Complete graph	$\mathcal{B}$	ILP		pars		PAUP*	
				length	time	length	time	length	time
<i>O. rufipogon</i> DNA	41 × 1043	$2^{18} * 3^2$	58	57	0.29s	57	2.57s	57	2.09s
Human mt-DNA	80 × 245	$2^{28}$	64	44	0.48s	45	0.56s	44	5.69s
HIV-1 RT protein	50 × 176	$2^{16} * 3 * 4^2$	297	40	127.5s	42	0.30s	40	3.84s
mt3000	500 × 3000	$2^{99} * 3^2$	322	177	40s	178	2m37s	177	5h23m
mt5000a	500 × 5000	$2^{167} * 3^2$	1180	298	5h10m	298	35m49s	298	3h52m
mt5000b	500 × 5000	$2^{229} * 3^3$	360	312	3m41s	312	57m6s	312	2h40m
mt10000	500 × 10000	$2^{357} * 3^3$	6006	N. A.	N. A.	637	1h34m	637	1h39m

### 3 Results

We implemented our generalized Buneman pruning and the ILP in C++. The ILP was solved using the Concert callable library of CPLEX 10.0. We compared the performance of our method with two popular heuristic methods for maximum parsimony phylogeny inference — **pars**, which is part of the freely-available PHYLIP package [13], and PAUP\* [12], the leading commercial phylogenetics package. We attempted to use PHYLIP’s exact branch-and-bound method DNA penny for nucleotide sequences, but discontinued the tests when it failed to solve any of the data sets in under 24 hours. In each case, **pars** and PAUP\* were run with default parameters. We first report results from three moderate-sized data sets selected to provide varying degrees of difficulty: a set of 1,043 sites from a set of 41 sequences of *O. rufipogon* (red rice) [10], 245 positions from a set of 80 human mt-DNA sequences reported by [11], and 176 positions from 50 HIV-1 reverse transcriptase amino acid sequences. The HIV sequences were retrieved by NCBI BLASTP [17] searching for the top 50 best aligned taxa for the query sequence GI 19571541 and default parameters. We then added additional tests on larger data sets all derived from human mitochondrial DNA. The mtDNA data was retrieved from NCBI BLASTN, searching for the 500 best aligned taxa for the query sequence GI 61287976 and default parameters. The complete set of 16,546 characters (after removing indels) was then broken in four windows of varying sizes and characteristics: the first 3,000 characters (mt3000), the first 5,000 characters (mt5000a), the next 5,000 characters (mt5000b), and the first 10,000 characters (mt10000). Table 1 summarizes the results.

For the set of 41 sequences of *lhc-1* gene from *O. rufipogon* (red rice) [10], our method pruned the full graph of  $2^{18} * 3^2$  nodes (after screening out redundant characters) to 58. Fig 4(a) shows the resulting phylogeny. Both PAUP\* and **pars** yielded an optimal tree although more slowly than the ILP (2.09 seconds and 2.57 seconds respectively, as opposed to 0.29 seconds).

For the 245-base human mt-DNA sequences, the generalized Buneman pruning was again highly efficient, reducing the state set from  $2^{28}$  after removing redundant sequences to 64. Fig 4(b) shows the phylogeny returned. While PAUP\* was able to find the optimal phylogeny (although it was again slower at 5.69



**Fig. 4.** Most parsimonious phylogenies (a) *lhs-1* gene for *O. rufipogon* [10] (b) Human mt-DNA [11] and (c) HIV-1 RT proteins [17]. Edges are labelled by their lengths in parentheses followed by sites that mutate along that edge. Dark red ovals are input taxa and light blue Steiner nodes.

seconds versus 0.48 seconds), **pars** yielded a slightly sub-optimal phylogeny (length 45 instead of 44) in a comparable run time (0.56 seconds).

For HIV-1 sequences, our method pruned the full graph of  $2^{16} * 3 * 4^2$  possible nodes to a generalized Buneman graph of 297 nodes, allowing solution of the ILP in about two minutes. Fig 4(c) shows an optimal phylogeny for the data. PAUP\* was again able to find the optimal phylogeny and in this case was faster than the ILP (3.84 seconds as opposed to 127.5 seconds). **pars** required a shorter run time of 0.30 seconds, but yielded a sub-optimal tree of length of 42, as opposed to the true minimum of 40.

For the four larger mitochondrial datasets, Buneman pruning was again highly effective in reducing graph size relative to the complete graph, although the ILP approach eventually proves impractical when Buneman graph sizes grows sufficiently large. Two of the data sets yielded Buneman graphs of size below 400, resulting in ILP solutions orders of magnitude faster than the heuristics. mt5000a, however, yielded a Buneman graph of over 1,000 nodes, resulting in an ILP that ran more slowly than the heuristics. mt10000 resulted in a Buneman graph of over 6,000 nodes, leading to an ILP too large to solve. **pars** was faster

than PAUP\* in all cases, but PAUP\* found optimal solutions for all three instances we can verify while **pars** found a sub-optimal solution in one instance.

We can thus conclude that the generalized Buneman pruning approach developed here is very effective at reducing problem size, but solving provably to optimality does eventually become impractical for large data sets. Heuristic approaches remain a practical necessity for such cases even though they cannot guarantee, and do not always deliver, optimality. Comparison of PAUP\* to **pars** and the ILP suggests that more aggressive sampling over possible solutions by the heuristics can lead optimality even on very difficult instances but at the cost of generally greatly increased run time on the easy to moderate instances.

## 4 Discussion

We have presented a new method for finding provably optimal maximum parsimony phylogenies on multi-state characters with weighted state transitions, using integer linear programming. The method builds on a novel generalization of the Buneman graph for characters with arbitrarily large but finite state sets and for arbitrary weight functions on character transitions. Although the method has an exponential worst-case performance, empirical results show that it is fast in practice and is a feasible alternative for data sets as large as a few hundred taxa. While there are many efficient heuristics for reconstructing maximum parsimony phylogenies, our results cater to the need for provably exact methods that are fast enough to solve the problem for biologically relevant multi-state data sets. Our work could potentially be extended to include more sophisticated integer programming techniques that have been successful in solving large instances of other hard optimization problems, for instance the recent solution of the 85,900-city traveling salesman problem `pla85900` [18]. The theoretical contributions of this paper may also prove useful to work on open problems in multi-state MP phylogenetics, to accelerating methods for related objectives, and to sampling among optimal or near-optimal solutions.

## Acknowledgements

NM would like to thank Ming-Chi Tsai for several useful discussions. This work was supported in part by NSF grant #0612099.

## References

1. Posada, D., Crandall, K.: Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology and Evolution* 16, 37–45 (2001)
2. Felsenstein, J.: *Inferring Phylogenies*. Sinauer Publications (2004)
3. Foulds, L.R., Graham, R.L.: The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics* 3, 43–49 (1982)
4. Sridhar, S., Lam, F., Belloch, G., Ravi, R., Schwartz, R.: Efficiently finding the most parsimonious phylogenetic tree via linear programming. In: Măndoiu, I.I., Zelikovsky, A. (eds.) *ISBRA 2007*. LNCS (LNBI), vol. 4463, pp. 37–48. Springer, Heidelberg (2007)



5. Buneman, P.: The recovery of trees from measures of dissimilarity. In: Hodson, F., et al. (eds.) *Mathematics in the archeological and historical sciences*, pp. 387–395 (1971)
6. Barthélemy, J.: From copair hypergraphs to median graphs with latent vertices. *Discrete Math.* 76, 9–28 (1989)
7. Bandelt, H.J., Forster, P., Sykes, B.C., Richards, M.B.: Mitochondrial portraits of human populations using median networks. *Genetics* 141, 743–753 (1989)
8. Bandelt, H.J., Forster, P., Rohl, A.: Median-joining networks for inferring intraspecific phylogenies. *Molecular Biology and Evolution* 16, 37–48 (1999)
9. Huber, K.T., Moulton, V.: The relation graph. *Discrete Mathematics* 244(1-3), 153–166 (2002)
10. Zhou, H.F., Zheng, X.M., Wei, R.X., Second, G., Vaughan, D.A., Ge, S.: Contrasting population genetic structure and gene flow between *Oryza rufipogon* and *Oryza nivara*. *Theor. Appl. Genet.* 117(7), 1181–1189 (2008)
11. Hudjashov, G., Kivisild, T., Underhill, P.A., Endicott, P., Sanchez, J.J., Lin, A.A., Shen, P., Oefner, P., Renfrew, C., Villems, R., Forster, P.: Revealing the prehistoric settlement of Australia by Y chromosome and mtDNA analysis. *Proc. Natl. Acad. Sci. USA* 104(21), 8726–8730 (2007)
12. Swofford, D.: PAUP\* 4.0. Sinauer Assoc. Inc., Sunderland (2009)
13. Felsenstein, J.: PHYLIP (phylogeny Inference package) version 3.6 distributed by author, Department of Genome Sciences, University of Washington, Seattle (2008)
14. Semple, C., Steel, M.: *Phylogenetics*. Oxford University Press, Oxford (2003)
15. Erdos, P.L., Szekely, L.A.: On weighted multiway cuts in trees. *Mathematical Programming* 65, 93–105 (1994)
16. Wang, L., Jiang, T., Lawler, L.: Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica* 16, 302–315 (1996)
17. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402 (1997)
18. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W., Espinoza, D.G., Goycoolea, M., Helsgaun, K.: Certification of an optimal TSP tour through 85,900 cities. *Operations Research Letters* 37(1), 11–15 (2009)

# Seed Design Framework for Mapping SOLiD Reads

Laurent Noé, Marta Gîrdea, and Gregory Kucherov\*

INRIA Lille - Nord Europe, LIFL/CNRS, Université Lille 1,  
59655 Villeneuve d'Ascq, France

**Abstract.** The advent of high-throughput sequencing technologies constituted a major advance in genomic studies, offering new prospects in a wide range of applications. We propose a rigorous and flexible algorithmic solution to mapping SOLiD color-space reads to a reference genome. The solution relies on an advanced method of seed design that uses a faithful probabilistic model of read matches and, on the other hand, a novel seeding principle especially adapted to read mapping. Our method can handle both lossy and lossless frameworks and is able to distinguish, at the level of seed design, between SNPs and reading errors. We illustrate our approach by several seed designs and demonstrate their efficiency.

## 1 Introduction

High-throughput sequencing technologies can produce hundreds of millions of DNA sequence reads in a single run, providing faster and less expensive solutions to a wide range of genomic problems. Among them, the popular SOLiD system (Applied Biosystems) features a 2-base encoding of reads, with an error-correcting capability helping to reduce the error rate and to better distinguish between sequencing errors and SNPs.

In this paper, we propose a rigorous and flexible algorithmic approach to mapping SOLiD color-space reads to a reference genome, capable to take into account various external parameters as well as intrinsic properties of reads resulting from the SOLiD technology. The flexibility and power of our approach comes from an advanced use of *spaced seeds* [12].

The main novelty of our method is an *advanced seed design* based on a *faithful probabilistic model of SOLiD read alignments* incorporating reading errors, SNPs and base indels, and, on the other hand, on a *new seeding principle* especially adapted for read mapping. The latter relies on the use of a small number of seeds (in practice, typically two) *designed simultaneously with a set of position on the read where they can hit*. We call this principle *position-restricted seeds*. Advantageously, it allows us to take into account, in a subtle way, read properties such as a non-uniform distribution of reading errors along the read, or a tendency of reading errors to occur periodically at a distance of 5 positions, which are observed artifacts of the SOLiD technology.

---

\* On leave in J.-V.Poncelet Lab, Moscow, Russia.

A number of algorithms and associated software programs for read mapping have been recently published. Several of them such as MAQ [3], MOSAIK [4], MPSCAN [5] PASS [6], PerM [7], RazerS [8], SHRiMP [9] or ZOOM [10] apply contiguous or spaced seeding techniques, requiring one or several hits per read. Other programs approach the problem differently, e.g., by using the Burrows-Wheeler transform (Bowtie [11], BWA [12], SOAP2 [13]), suffix arrays (segemehl [14], BFAST [15]), variations of the Rabin-Karp algorithm (SOCS [16]) or a non-deterministic automata matching algorithm on a keyword tree of the search strings (PatMaN [17]). Some tools, such as segemehl [14] or Eland [18], are designed for 454 and Illumina reads and thus do not deal with the characteristics of the SOLiD encoding which is the subject of this paper. Also, it should be noted that, in many cases, sensitivity is sacrificed in favor of speed: most methods find similarities up to a small number of mismatches, and few approaches account for nucleotide insertions and deletions.

Seed-based methods for read mapping use different seeding strategies. SHRiMP [9] uses spaced seeds that can hit at any position of the read and introduces a lower bound on the number of hits within one read. MAQ [3] uses six light-weight seeds allowed to hit in the initial part of the read. ZOOM [10] proposes to use a small number (4-6) of spaced seeds each applying at a fixed position, to ensure a lossless search with respect to a given number of mismatches. In the lossless framework, PerM [7] proposes to use “periodic seeds” (see also [19]) to save on the index size.

Despite the number of proposed solutions, none of them relies on a systematic seed design method taking into account (other than very empirically) statistical properties of reads. In this paper, we present a seed design based on Hidden Markov models of read matches, using a formal finite automata-based approach previously developed in [20]. To the best of our knowledge, this is the first time that the seed design for read mapping is done based on a rigorous probabilistic modeling.

Our approach allows us to design seeds in both lossy and lossless frameworks. In the lossless framework, where the goal is to detect all read occurrences within a specified number of mismatches, we have the flexibility of partitioning this number into reading errors and SNPs.

As a result, we obtain a very efficient mapping algorithm combining a small number of seeds and therefore a reasonable amount of index memory with guaranteed sensitivity and small running time, due to a restricted subset of positions where seeds should be applied.

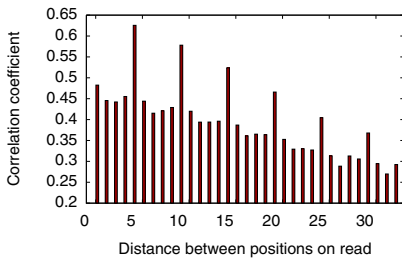
## 2 AB SOLiD Reads: Encoding and Technological Artifacts

The SOLiD System [21] enables massively parallel sequencing of clonally amplified DNA fragments. This sequencing technology is based on sequential ligation of dye-labeled oligonucleotide probes, each probe assaying two base positions at a time. The system uses four fluorescent dyes to encode for the sixteen possible 2-base combinations. Consequently, a DNA fragment is represented by the

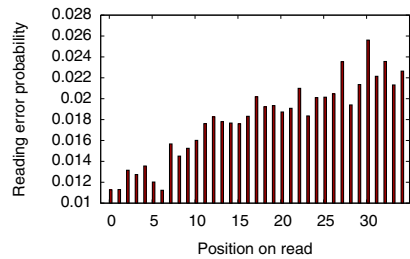
initial base followed by a sequence of overlapping dimers, each encoded with one of four colors using a degenerate coding scheme that satisfies several rules. Thus, although a single color in a read can represent any of four dimers, the overlapping properties of the dimers and the nature of the color code eliminate ambiguities and allow for error-correcting properties.

As our work relies on modeling the error distribution along the reads, we are particularly interested in several aspects of the sequencing technology that influence this distribution. First, since every color of the read encodes two adjacent bases and therefore every base affects two adjacent colors, it follows that any single base mutation results in the change of two adjacent colors in the read. On the other hand, since cycles of five di-nucleotide readings are performed in order to retrieve the sequence (as described in the documentation of Applied Biosystems [21][22]), we expect reading error bias to appear with a periodicity of 5.

To confirm this intuition, we studied the variation of the reading error probability along the read by analyzing statistical properties of about a million of SOLiD reads of the *S. cerevisiae* genome. In this analysis, we used the qualities  $Q_l$  associated to each position  $l$  on the read, which relate to the error probability  $p_e^l$  through  $Q_l = -10 \cdot \log_{10}(p_e^l)$  [23].



**Fig. 1.** Position quality correlation coefficient depending on the distance between read positions



**Fig. 2.** Average reading error probability at each read position

We computed the quality correlation between read positions depending on the distance between them. Formally, if  $m$  is the read length, then for each  $i \in \{1, \dots, m-1\}$ , we computed the correlation through the following standard formula  $c(i) = \frac{E((Q_j - \tilde{Q})(Q_{j+i} - \tilde{Q}))}{(\sigma_Q)^2}$ , where  $E(\cdot)$  is the expectation,  $\tilde{Q}$  the average quality along the read, and  $\sigma_Q$  the standard deviation of quality values. The result is given in Figure 1. It shows significantly higher correlations (up to 0.63) between pairs of positions located at distances that are multiples of 5.

Additionally, we studied the behavior of reading error probability values along the read. As shown in Figure 2 the error probability tends to increase towards the end of the read, making the last positions of the color sequence less reliable when searching for similarities.

## 3 Seed Design for Mapping SOLiD Reads

### 3.1 Seed Design: Background

Spaced seeds, first proposed in the context of DNA sequence alignment by the PatternHunter algorithm [1], represent a powerful tool for enhancing the efficiency of the sequence search.

Using a spaced seed instead of a contiguous stretch of identical nucleotides to select a potential similarity region can improve the sensitivity of the search for a given selectivity level [1]. Furthermore, using a seed family, i.e. several seeds simultaneously instead of a single seed, further improves the sensibility/selectivity trade-off [24,25]. The price for using seed families is the necessity to store in memory several indexes, one for each seed. In practice, however, using in the search a small number of seeds can significantly improve the sensitivity/selectivity ratio.

A crucial feature of spaced seeds is their capacity to be adapted to different search situations. Spaced seeds can be *designed* to capture statistical properties of sequences to be searched. For example, [26,27] report on designing spaced seeds adapted to the search of coding regions. One of the contributions of this paper is a rigorous design of seeds adapted to mapping genomic reads issued from the SOLiD technology. Note that here we will work with regular spaced seeds rather than more advanced subset seeds [20,27,28], as there is very little or no information in discriminating among different classes of mismatches that can be used to our advantage.

One has to distinguish between the *lossy* and *lossless* cases of seed-based search. In the lossy case we are allowed to miss a fraction of target matches, and the usual goal of seed design is to maximize the sensitivity over a class of seeds verifying a certain selectivity level. In the lossless case we must detect all matches verifying a given dissimilarity threshold (expressed in terms of a number of errors or a minimal score), and the goal of seed design is to compute a minimal set of seeds with the best selectivity that still ensures the lossless search. In the context of read mapping for high-throughput sequencing technologies, both lossy [9,3] and lossless [10,7] frameworks have been used.

Our approach to seed design relies on a methodology proposed in our previous work [20], based on the finite automata theory. A central idea is to model the set of target alignments by a *finite-state probability transducer*, which subsumes the Hidden Markov Model commonly used in biosequence analysis. On the other hand, a seed, or a seed family, is modeled by a *seed automaton* for which we proposed an efficient compact construction [29]. Once these two automata have been specified, computing the seed sensitivity can be done efficiently with a dynamic programming algorithm as described in [20]. The seed design is then done by applying our IEDERA software [20,29,30] that uses the above algorithm to explore the space of possible seeds and select most sensitive seeds using a sampling procedure for seeds and respective hit positions and by performing a local optimization on the best candidates.

Here we apply this methodology to seed design for mapping SOLiD reads, both in the lossy and lossless frameworks. Besides, we introduce an important

novelty in the definition of seeds, especially advantageous for mapping short reads: *position-restricted seeds*, which are seeds designed together with the set of positions on the read where they can be applied. This can be seen as an intermediate paradigm between applying each seed at every position and the framework of [10] where each seed applies to a designated position of the read. Position-restricted seeds offer an additional power of capturing certain read properties (such as, e.g., an increasing error level towards the end of the read) in a flexible way, without sacrificing the selectivity and thus the speed of the seeding procedure.

### 3.2 Modeling Seeds and SOLiD Reads by Finite Automata

We now present our model of color sequence alignments, built on the observations of Section 2. Note that we consider the reference genome translated into the color alphabet, i.e. both the reads and the genome are represented in color space.

**Position-restricted seeds.** As shown in Section 2, the reading error probability increases towards the end of the read, implying that a search for similarity within the last positions of the read could lead to erroneous results or no results at all. Hence, we can improve the seed selectivity by favoring hits at initial positions of the read where matches are more likely to be significant. We then define each seed  $\pi$  *jointly* with a set of positions  $P$  to which it is applied on the read.

We use the framework of [20] where a seed  $\pi$  is represented by a deterministic finite automaton  $\mathcal{Q}$  over the alignment alphabet  $\mathcal{A}$  which is here the binary match/mismatch alphabet. Note that the size of  $\mathcal{Q}$  is a crucial parameter in the algorithm of [20] to compute the sensitivity of the seed. An efficient construction of such an automaton has been studied in [29]: it has the optimal size of  $(w + 1)2^{s-w}$  states, where  $s$  and  $w$  are respectively the *span* (length) and *weight* (number of *match* symbols) of the seed.

Let  $m$  be the read size. To take into account the set of allowed positions, we compute the product of  $\mathcal{Q}$  with an automaton  $\lambda_P$  consisting of a linear chain of  $m + 1$  states  $q_0, q_1, \dots, q_m$ , where  $q_0$  is the initial state, and for every  $q_i$ , both outgoing transitions lead to  $q_{i+1}$ . Final states of the automaton reflect the set of possible positions  $P$  where the seed is allowed to hit: a state  $q_i$  is final iff  $i - s \in P$ .

A trivial upper bound on the size of the product automaton for a spaced seed of span  $s$  and weight  $w$  is  $(w + 1) \cdot 2^{s-w} \cdot m$ . This bound can be improved using the notion of matching prefix, as explained in [29]. Thus, an economical implementation of the product of  $\mathcal{Q}$  by  $\lambda$  taking into account the set of matching positions  $P$  always produces at most  $(w + 1) \cdot 2^{s-w} \cdot |P| + m$  states.

Furthermore, consider an interval graph of the possible placements of the seed on the read, where each placement spans over an interval of  $s$  positions. The chromatic number  $c$  of this graph can be easily computed, providing the maximal number of overlapping seeds. We observe that if this number is small (compared to  $(s - w + \log(w))$ ), then the size of the product automaton is bounded by  $O((m + 1) \cdot 2^c)$ .

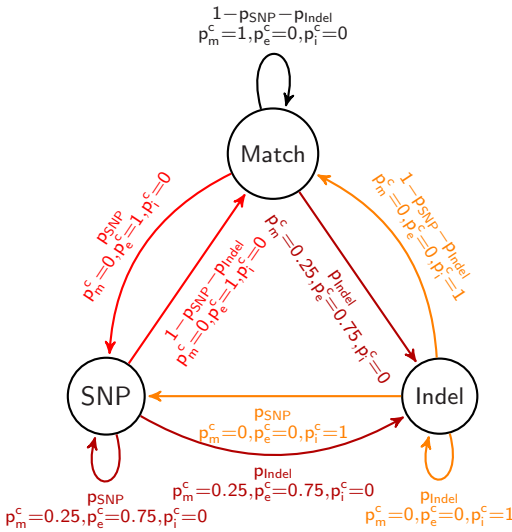
**Model for SNPs and reading errors.** As explained in Section 2, there are two independent sources of errors in reads with respect to the reference genome: reading errors and SNPs/indels, i.e., *bona fide* differences between the reference genome and sequenced data. We represent each of these sources by a separate Hidden Markov Model (viewed as a probabilistic transducer, see [20]), combined in a model which allows all error types to be cumulated in the resulting sequences.

The **SNP/Indel model**, denoted  $M_{SNP/I}$ , (Figure 3) has three states: *Match*, *SNP* and *Indel*, referring to matches, mismatches, and indels at the nucleotide level, and is parametrized by SNP and Indel occurrence probabilities, denoted  $p_{SNP}$  and  $p_{Indel}$ . Each transition of  $M_{SNP/I}$  generates a *color match*, *mismatch* or *indel*, with probabilities  $p_m^c$ ,  $p_e^c$ , and  $p_i^c$  respectively, defined as follows. An insertion or deletion of  $n$  nucleotides appears at the color level as an insertion/deletion of  $n$  colors preceded in 3/4 cases by a color mismatch [21]. Hence, the  $p_e^c = 0.75$  when entering the *Indel* state, and  $p_i^c = 1$  for any transition having the *Indel* state as source. A nucleotide mutation is reflected in the color encoding by a change of two adjacent colors (and, more generally,  $n$  consecutive mutations affect  $n + 1$  consecutive colors [21]). Thus,  $p_e^c = 1$  when entering or leaving the *SNP* state, and a color match/mismatch mixture when staying in the mismatch state, since color matches may occur inside stretches of consecutive SNPs. Finally,  $p_m^c = 1$  when looping on the *M* state.

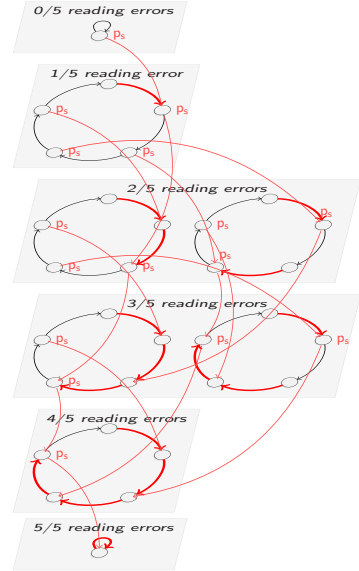
The **reading errors** are handled by a more complex model, denoted  $M_{RE}$  (Figure 4). Basically, it is composed of several submodels, one for each possible arrangement of reading errors on a cycle of 5 positions. Within these submodels, the transitions shown in red correspond to periodic reading errors, and generate reading errors with a fixed, usually high probability  $p_{err}$ . This simulates the periodicity property shown in Figure 1. Switching from one cyclic submodel to another with a higher reading error rate (by adding another red transition, with high error probability) can occur at any moment with a fixed probability  $p_s$ .

The transitions shown in black in the model from Figure 4 have an error emission probability of 0. However, in the complete reading error model, we wish to simulate the error probability that increases towards the end (in conformity with Figure 2). We do this by ensuring that reading errors are generated on these transitions with a probability  $p'_{err}(pos)$  (lower than  $p_{err}$ ) given by an increasing function of the current position  $pos$  on the read. Technically, this is achieved by multiplying the automaton in Figure 4 by a linear automaton with  $m + 1$  states, where  $m$  is the read length and the  $i$ -th transition generates a reading error (color mismatch) with the probability  $p'_{err}(i)$ . The reading error emission probability in the product model is computed as the maximum of the two reading error probabilities encountered in the multiplied models.

The **final model**, which combines both error sources, is the product of  $M_{SNP/I}$  and  $M_{RE}$ . While the states and transitions of the product model are defined in the classic manner, the emissions are defined through specific rules based on symbol priorities. If corresponding transitions of  $M_{SNP/I}$  and  $M_{RE}$  generate symbols  $\alpha$  and  $\beta$  with probabilities  $p_1$  and  $p_2$  respectively, then the product automaton



**Fig. 3.** Model of SNPs and Indels ( $M_{SNP/I}$ ). Colors of transitions correspond to emitted errors: black for color matches, red for mismatches, yellow for indels, and dark red for a mixture of matches (0.25) and mismatches (0.75).



**Fig. 4.** Reading error automaton

generates the dominant symbol between  $\alpha$  and  $\beta$  with probability  $p_1 p_2$ . Different probabilities obtained in this way for the same symbol are added up.

The dominance relation is defined as follows: *indels* are dominant over both *mismatches* and *matches*, and *mismatches* dominate *matches*. For example, (*indel*, *mismatch*) results in an *indel*, (*mismatch*, *mismatch*) and (*match*, *mismatch*) represent *mismatch*, (*match*, *match*) is a *match*. This approach ensures that errors generated by each of the two models are superposed.

### 3.3 Computing the Sensitivity or Testing the Lossless Property

Given an automaton  $\mathcal{Q}$  specifying a family of seeds possibly restricted to a set of positions, we have to compute its sensitivity (in the lossy framework) or to test whether it is lossless (in the lossless framework).

The sensitivity of a seed family is defined [131] as the probability for at least one of the seeds to hit a read alignment with respect to a given probabilistic model of the alignment. As outlined in Section 3.1, this is done using the dynamic programming technique of [20]. We therefore omit further details.

In the lossless framework, we have to test if the seed specified by  $\mathcal{Q}$  is lossless, i.e. hits *all* the target alignments. The set of target alignments is defined through a threshold number of allowed mismatches.

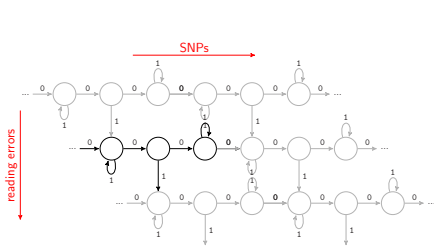


A straightforward way to test the lossless property of  $\mathcal{Q}$  would be to construct a deterministic automaton recognizing the set of all target alignments and then to test if the language of this automaton is included in the language of  $\mathcal{Q}$ . This, however, is unfeasible in practice. The automaton of all target alignments is much too costly to construct: for example, in the case of threshold of  $k$  mismatches, there are  $\sum_{a=0}^k \binom{m}{a}$  different alignments of length  $m$ , and the Aho-Corasick automaton of these strings would have  $\sum_{a=0}^{k+1} \binom{m}{a}$  states. Moreover, testing the inclusion would lead to computing the product of this automaton with  $\mathcal{Q}$  which would multiply the number of states by that of  $\mathcal{Q}$ .

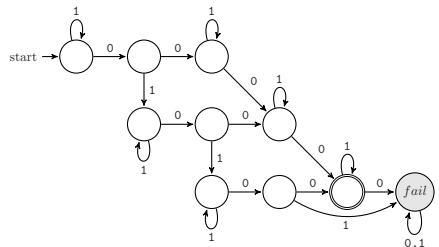
Alternatively, we propose an efficient dynamic programming algorithm directly applied to  $\mathcal{Q}$  that can verify the inclusion. This algorithm computes, for each state  $q$  of  $\mathcal{Q}$ , and for each iteration  $i \in [1..m]$ , the minimal number of mismatches needed to reach  $q$  at step  $i$ . Let  $k$  be the threshold for the number of mismatches. Then, the lossless condition holds iff at step  $m$ , all non-final states have a number of mismatches greater than  $k$ . Indeed, if there is a non-final state that has a number of errors at most  $k$  after  $m$  steps, then there is at least one string of length  $m$  with at most  $k$  mismatches that is not detected by the automaton, which contradicts the lossless condition. This algorithm is of time complexity  $\mathcal{O}(|\mathcal{Q}| \cdot |\mathcal{A}| \cdot m)$ , and space complexity  $\mathcal{O}(|\mathcal{Q}| \cdot |\mathcal{A}|)$ , where  $\mathcal{A}$  is the alphabet of the alignment sequences, in our case  $\{0, 1\}$ .

To illustrate the efficiency of this algorithm, consider the case of a single spaced seed of span  $s$  and weight  $w$ , yielding an automaton with at most  $(w + 1) \cdot 2^{s-w}$  states [32,20]. On this automaton, our method runs in time  $\mathcal{O}(wm2^{s-w})$  which brings an improvement by a factor of  $\frac{2^w}{w}$  of the general bound  $\mathcal{O}(m2^s)$  from [33].

In the context of color sequence mapping, it is interesting to define the lossless property with respect to a *maximal number of allowed mismatches that is split between SNPs and reading errors*. Since, in the color space, a SNP appears as two adjacent color mismatches, having  $k$  non-consecutive SNPs and  $h$  color mismatches implies the possibility to accept  $2k + h$  mismatches with the additional restriction that there exist at least  $k$  pairs of adjacent ones. The automaton that recognizes the set of alignments verifying this condition on mismatches can be obtained by combining simple 3-state building blocks as depicted in Figure 5.



**Fig. 5.** Building an automaton for  $k$  SNPs and  $h$  color mismatches from a repeated 3-state pattern



**Fig. 6.** 1 SNP & 2 errors automaton



be a consequence of the fact that we consider indels in our lossy model, which usually forces the seeds to have a smaller span. Another interesting observation is that two-seed families 2-LOSSY-10P and 2-LOSSY-12P are actually lossless for the threshold of 3 mismatches, whereas single seeds 1-LOSSY-10P and 1-LOSSY-12P are not lossless for this setting.

We then focused on the lossless case where the maximal number of allowed mismatches is split between SNPs and reading errors. Using the procedure described in Section 3.3, we computed lossless single and double seeds for one SNP and two reading errors. Results are shown in Figure 8.

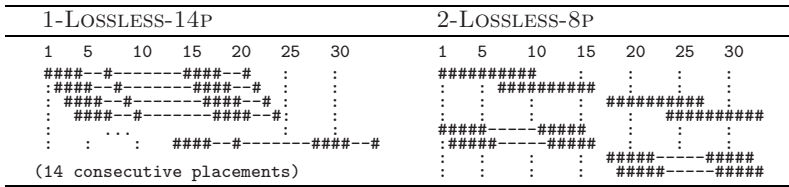


Fig. 8. Lossless position-restricted seeds for 1 SNP and 2 reading errors

Note that the seed 1-LOSSLESS-14P is one of several single seeds of weight 10 we found that satisfied this lossless condition, with no restriction on allowed positions. Interestingly, they all have a very large span (21) and a regular pattern with a periodic structure that can be obtained by iterating a simpler pattern solving the lossless problem for an appropriate *cyclic problem*, following the property we previously described in [19]. For two-seed families, Figure 8 shows a lossless pair of seeds 2-LOSSLESS-8P for read length 33 (which then remains lossless for larger lengths), where each seed is restricted to apply to four positions only.

To get a better idea of the sensitivity of the obtained seeds applied to real data, we tested them on 100000 reads of length 34 from *S. cerevisiae* and computed the number of read/reference alignments hit by each (single or double) seed. Alignments were defined through the score varying from 28 to 34, under the scoring scheme +1 for match, 0 for color mismatch or SNP, -2 for gaps. Results are presented in Figure 9. One conclusion we can draw is that the performance of lossless seeds 1-LOSSLESS-14P and 2-LOSSLESS-8P decreases quite fast when the alignment score goes down, compared to lossy seeds. Intuitively, this is, in a sense, a price to pay for the lossless condition which usually makes these seeds less appropriate for the alignments with a number of errors exceeding the threshold. Another conclusion is that, as expected, single seeds perform worse than double seeds, although the overall number of positions where seeds apply is the same for both single and double seeds.

Note finally that the choice of the best seed can be affected, on one hand, by different properties of the class of target alignments (number, type and distribution of mismatches and indels etc.) and, on the other hand, by the size of the data and the available computational resources. The former can be captured

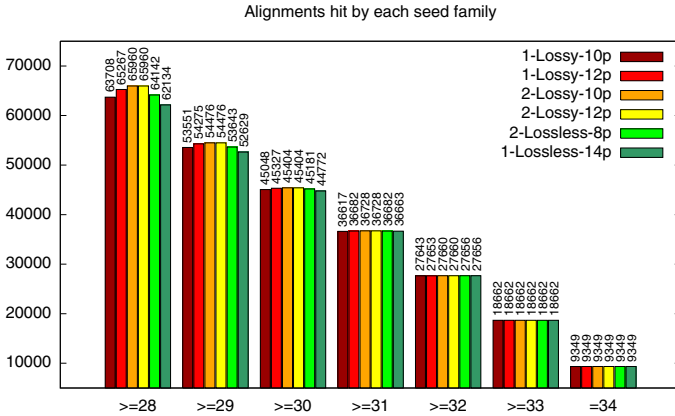


Fig. 9. Number of read alignments with scores between 28 and 34 hit by each seed

by our probabilistic models described in Section 3. The latter is related to the choice of the selectivity level, directly affecting the speed of the search, which is defined by the seed weight and the number of allowed positions. Depending on the chosen selectivity, different seeds can (and should) be preferred. Note in this regard that seeds appearing in Figure 9 have different selectivity and are then incomparable *stricto sensu*. A comparison of different seeds for SOLiD read mapping in typical practical situations will be a subject of a separate work.

## 5 Conclusions and Perspectives

In this paper, we presented a seed design framework for mapping SOLiD reads to a reference genomic sequence. Our contributions include the concept of position-restricted seeds, particularly suitable for short alignments with non-uniform error distribution; a model that captures the statistical characteristics of the SOLiD reads, used for the evaluation of lossy seeds; an efficient dynamic programming algorithm for verifying the lossless property of seeds; the ability to distinguish between SNPs and reading errors in seed design.

Our further work will include a more rigorous training of our models and in particular a more accurate estimation of involved probabilities, possibly using advanced methods of assessing the fit of a model. Another interesting question to study is the design of efficient combined lossy/lossless seeds which provide a guarantee to hit all the alignments with a specified number of errors and still have a good sensitivity when this threshold is exceeded. Computing such seeds, however, could be difficult or even unfeasible: for example, lossless seeds tend to have a regular structure (see [19]) while best lossy seeds often have asymmetric and irregular structure. Finally, we want to define and study a lossless property that incorporates possible indels and not only mismatches (SNPs or reading errors) occurring in read alignments.

## Acknowledgments

The authors would like to thank Valentina Boeva and Emmanuel Barillot from the *Institut Marie Curie* at Paris for helpful discussions and for providing the dataset of *Saccharomyces cerevisiae* reads that we used as a testset in our study. We also thank Martin Figeac (*Institut national de la santé et de la recherche médicale*) for sharing insightful knowledge about the SOLiD technology. Laurent Noé was supported by the ANR project CoCoGen (BLAN07-1 185484).

## References

1. Ma, B., Tromp, J., Li, M.: PatternHunter: Faster and more sensitive homology search. *Bioinformatics* 18(3), 440–445 (2002)
2. Noé, L., Kucherov, G.: YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Research* 33(Web Server Issue), W540–W543 (2005)
3. Li, H., Ruan, J., Durbin, R.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18, 1851–1858 (2008)
4. Strömberg, M., Lee, W.P.: MOSAIK read alignment and assembly program (2009), <http://bioinformatics.bc.edu/marthlab/Mosaik>
5. Rivals, E., Salmela, L., Kiiskinen, P., Kalsi, P., Tarhio, J.: MPSCAN: Fast localisation of multiple reads in genomes. In: Salzberg, S.L., Warnow, T. (eds.) *Algorithms in Bioinformatics*. LNCS, vol. 5724, pp. 246–260. Springer, Heidelberg (2009)
6. Campagna, D., Albiero, A., Bilardi, A., Caniato, E., Forcato, C., Manavski, S., Vitulo, N., Valle, G.: PASS: a program to align short sequences. *Bioinformatics* 25(7), 967–968 (2009)
7. Chen, Y., Souaiaia, T., Chen, T.: PerM: Efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics* 25(19), 2514–2521 (2009)
8. Weese, D., Emde, A., Rausch, T., Döring, A., Reinert, K.: RazerS—fast read mapping with sensitivity control. *Genome Research* 19(9), 1646–1654 (2009)
9. Rumble, S.M., Lacroute, P., Dalca, A.V., Fiume, M., Sidow, A., Brudno, M.: SHRiMP: Accurate mapping of short color-space reads. *PLoS Comp. Biol.* 5(5) (2009)
10. Lin, H., Zhang, Z., Zhang, M., Ma, B., Li, M.: ZOOM! zillions of oligos mapped. *Bioinformatics* 24(21), 2431–2437 (2008)
11. Langmead, B., Trapnell, C., Pop, M., Salzberg, S.: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10(3) (2009)
12. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25(14), 1754–1760 (2009)
13. Li, R., Yu, C., Li, Y., Lam, T., Yiu, S., Kristiansen, K., Wang, J.: SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25(15), 1966–1967 (2009)
14. Hoffmann, S., Otto, C., Kurtz, S., Sharma, C., Khaitovich, P., Stadler, P., Hackermüller, J.: Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comp. Biol.* 5(9) (2009)
15. Homer, N., Merriman, B., Nelson, S.F.: BFAST: an alignment tool for large scale genome resequencing. *PLoS One* 4(11) (2009)

16. Ondov, B., Varadarajan, A., Passalacqua, K., Bergman, N.: Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications. *Bioinformatics* 24(23), 2776–2777 (2008)
17. Pruffer, K., Stenzel, U., Dannemann, M., Green, R., Lachmann, M., Kelso, J.: PatMaN: rapid alignment of short sequences to large databases. *Bioinformatics* 24(13), 1530–1531 (2008)
18. Bentley, D., Balasubramanian, S., Swerdlow, H., Smith, G., Milton, J., Brown, C., Hall, K., Evers, D., Barnes, C., Bignell, H., et al.: Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* 456(7218), 53–59 (2008)
19. Kucherov, G., Noé, L., Roytberg, M.: Multiseed lossless filtration. *IEEE Transactions on Computational Biology and Bioinformatics* 2(1), 51–61 (2005)
20. Kucherov, G., Noé, L., Roytberg, M.: A unifying framework for seed sensitivity and its application to subset seeds. *J. Bioinform. Comput. Biol.* 4(2), 553–570 (2006)
21. ABI: A theoretical understanding of 2 base color codes and its application to annotation, error detection, and error correction. methods for annotating 2 base color encoded reads in the SOLiD™ system (2008)
22. ABI: The SOLiD™3 system. enabling the Next Generation of Science (2009)
23. Ewing, B., Green, P.: Base-calling of automated sequencer traces using phred. II. error probabilities. *Genome Research* 8(3), 186–194 (1998)
24. Li, M., Ma, B., Kisman, D., Tromp, J.: PatternHunter II: Highly sensitive and fast homology search. *J. Bioinform. Comput. Biol.* 2(3), 417–439 (2004)
25. Sun, Y., Buhler, J.: Designing multiple simultaneous seeds for DNA similarity search. *Journal of Computational Biology* 12(6), 847–861 (2005)
26. Brejová, B., Brown, D.G., Vinar, T.: Optimal spaced seeds for hidden markov models, with application to homologous coding regions. In: Baeza-Yates, R., Chávez, E., Crochemore, M. (eds.) *CPM 2003*. LNCS, vol. 2676, pp. 42–54. Springer, Heidelberg (2003)
27. Zhou, L., Stanton, J., Florea, L.: Universal seeds for cDNA-to-genome comparison. *BMC Bioinformatics* 9(36) (2008)
28. Yang, J., Zhang, L.: Run probabilities of seed-like patterns and identifying good transition seeds. *Journal of Computational Biology* 15(10), 1295–1313 (2008)
29. Kucherov, G., Noé, L., Roytberg, M.: Subset seed automaton. In: Holub, J., Žďárek, J. (eds.) *CIAA 2007*. LNCS, vol. 4783, pp. 180–191. Springer, Heidelberg (2007)
30. Kucherov, G., Noé, L., Roytberg, M.: Iedera: subset seed design tool (2009), <http://bioinfo.lifl.fr/yass/iedera>
31. Keich, U., Li, M., Ma, B., Tromp, J.: On spaced seeds for similarity search. *Discrete Applied Mathematics* 138(3), 253–263 (2004); preliminary version in 2002
32. Buhler, J., Keich, U., Sun, Y.: Designing seeds for similarity search in genomic DNA. In: *Proceedings of the 7th Annual International Conference on Computational Molecular Biology (RECOMB)*, pp. 67–75. ACM Press, New York (2003)
33. Burkhardt, S., Kärkkäinen, J.: Better filtering with gapped  $q$ -grams. *CPM 2001* 56(1,2), 51–70 (2003); Preliminary version in *CPM 2001*

# Accurate Estimation of Expression Levels of Homologous Genes in RNA-seq Experiments

Bogdan Paşaniuc<sup>1</sup>, Noah Zaitlen<sup>2,3</sup>, and Eran Halperin<sup>1,2,3</sup>

<sup>1</sup> International Computer Science Institute, Berkeley, CA

<sup>2</sup> Molecular Microbiology and Biotechnology Department  
Tel-Aviv University

<sup>3</sup> The Blavatnik School of Computer Science  
Tel-Aviv University

**Abstract.** Next generation high throughput sequencing (NGS) is poised to replace array based technologies as the experiment of choice for measuring RNA expression levels. Several groups have demonstrated the power of this new approach (RNA-seq), making significant and novel contributions and simultaneously proposing methodologies for the analysis of RNA-seq data. In a typical experiment, millions of short sequences (reads) are sampled from RNA extracts and mapped back to a reference genome. The number of reads mapping to each gene is used as proxy for its corresponding RNA concentration. A significant challenge in analyzing RNA expression of homologous genes is the large fraction of the reads that map to multiple locations in the reference genome. Currently, these reads are either dropped from the analysis, or a naïve algorithm is used to estimate their underlying distribution. In this work, we present a rigorous alternative for handling the reads generated in an RNA-seq experiment within a probabilistic model for RNA-seq data; we develop maximum likelihood based methods for estimating the model parameters. In contrast to previous methods, our model takes into account the fact that the DNA of the sequenced individual is not a perfect copy of the reference sequence. We show with both simulated and real RNA-seq data that our new method improves the accuracy and power of RNA-seq experiments.

## 1 Introduction

Next generation high throughput sequencing (NGS) technologies are rapidly establishing themselves as powerful tools for assaying a growing list of cellular properties including sequence and structural variation, RNA expression levels, alternative splice variants, protein-DNA/RNA interaction sites, and chromatin methylation state [18,16,14,15,11]. NGS enables thousands of megabases of DNA to be sequenced in a matter of days with very low cost compared to traditional Sanger sequencing. It provides tens of millions of short reads which can then be mapped back to a reference genome or used for de novo assembly. The advantages offered by NGS are underlined by the sheer wealth of significant

novel discoveries not possible with existing chips and prohibitively expensive with previous sequencing methods.

As with any new technology, there are a host of new problems to solve in order to maximize the benefit of the data produced. In the case of NGS, many of the new methods adapt classic problems such as alignment and assembly to the relatively short, inaccurate, and abundant set of reads. Other methods, such as the one presented here, aim at optimizing the analysis of NGS assays previously done using microarray based technologies such as quantifying gene expression levels from RNA data (*RNA-seq*). A first step in such an analysis is mapping the reads to a reference genome and aggregating the counts for each genomic location. Under the assumption that NGS samples short reads at random from the sequenced sample, the sequences with higher concentration will produce more reads. In the case of arrays this corresponds to a higher probe intensity. Indeed, it was recently shown that the RNA-seq read counts and expression array probe intensities are highly correlated measurements for RNA expression levels [15, 14].

Accurate estimation of the number of reads mapped to each genomic location critically depends on finding the location on the reference genome from which each read originated. While the majority of the reads produced by an NGS experiment map to a unique location along the genome, due to short read length, sequencing errors, and the presence of repetitive elements and homologs, a significant percentage of reads (up to 30% from the total mappable reads) are mapped to multiple locations (*multireads*). In the vast majority of RNA-seq experiments that have been published so far, the analysis consisted of simply disregarding the multireads from subsequent analyses. However, as previously noticed [15] if the multireads are discarded, the expression levels of genes with homologous sequences will be artificially deflated. If the multireads are split randomly amongst their possible loci, differences in estimates of expression levels for these genes between conditions will also be diminished leading to lower power to detect differential gene expression. Several groups have proposed a more intuitive alternative for dealing with multireads [6, 15]. Although there are small differences, they both adopt a heuristic approach, dividing the multireads amongst their mapped regions according to the distribution of the uniquely mapped reads in those regions. Intuitively, if there is a unique segment in the homologous region, then the distribution of the multireads in the repetitive segment of the region will follow the same distribution as the reads in the unique segment. This approach, although intuitive, is not optimal, as it does not thoroughly model the contribution of the multireads.

In this work we propose a rigorous framework for handling multireads that is applicable to several different assays including RNA-seq. In contrast to previous approaches, which were heuristic in their nature, we propose a generative model that describes the results of an RNA-seq experiment including multireads. An important feature of our model is that it takes into account genetic variation between the reference human genome sequence and the sequence of the studied sample, improving accuracy in some instances and allowing for simultaneous expression analysis and genotyping. We further developed algorithms for



estimating the parameters of the model using a maximum likelihood approach. We show through simulations and real RNA-seq data that our method significantly improves the accuracy and power of detecting differentially expressed genes under several measures. Particularly, our results on real data demonstrate that in an RNA-seq experiment comparing two tissues, we can potentially discover many more genes that are differently expressed between the tissues. In addition, our treatment of genetic variation allows us to simultaneously call variants (e.g. locations where the sequenced sample varies from reference), and use the location of these variants to further resolve the location of the multireads.

An implementation of our method is freely available for download as part of the software package SeqEm at <http://www.cs.tau.ac.il/~heran/cozygene/software.html>.

## 2 Methods

We will first describe our probabilistic generative model for an RNA-seq experiment. Let  $G = (G_1, \dots, G_n)$  be  $n$  contiguous DNA regions representing genes or other potentially expressed sequences. For each  $G_i$  we define the RNA cellular concentration of the gene as  $P_i$ , s.t.  $\sum_{i=1}^n P_i = 1$ .  $P = (P_1, \dots, P_n)$  can be interpreted as the normalized expression levels for the regions in  $G$ . Our model assumes that reads of length  $l$  are generated by randomly picking a region  $R$  from  $G$  according to the distribution  $P$ , and then copying  $l$  consecutive positions from  $R$  starting at a random position in the gene. The copying process is error-prone, with probability  $\epsilon(k)$  for a sequencing error in the  $k^{th}$  position of the read. The model is easily adapted to multi-length reads, but a fixed length is used here for simplicity. This process is repeated until we have a set of  $m$  reads  $R = r_1, \dots, r_m$  generated according to the model described above. The objective of an RNA-seq experiment is to infer  $P$  from  $R$ .

The first step in an RNA-seq experiment consists of mapping the results of an NGS run to the reference genome. Mapping methods such as ELAND, Maq, and bwa [9,12,13] provide for each read its most probable alignment, its position, and how many mismatches the alignment contains. Due to sequencing errors, some reads may not align perfectly. Furthermore, multireads align to more than one position, especially if the sequenced regions overlap with repeated genomic sequences such as homologous genes or repeats like ALUs, LINES, and SINES.

In the context of our model, each read  $r_i$  originated from one of the regions in  $G$ , but due to sequencing errors it may not align perfectly to that region; furthermore due to repeated sequences, it may also align to other regions. Put differently, for each region  $G_j$  and read  $r_i$ , we have a probability  $p_{ij} = P(r_j|G_i)$ , the probability of observing  $r_j$  given that the locus of the read was gene  $G_i$ . In practice, for each read  $r_j$ , this probability will be close to zero for all but a few regions. The likelihood of observing the  $m$  reads can be written as:

$$L(P; R) = \prod_{j=1}^m P(r_j|G, P) = \prod_{j=1}^m \sum_{i=1}^n P(G_i)P(r_j|G_i) = \prod_{j=1}^m \sum_{i=1}^n P_i p_{ij}$$

Unfortunately we do not know the expression levels  $P$ . A natural way of finding estimates for  $P$  is given in the following problem formulation for the Maximum Likelihood Expression Inference (MLEI) problem:

**Definition 1 (MLEI).** *Given a set of reads  $r_1, \dots, r_m$  and a set of regions  $G_1, \dots, G_n$ , find a probability  $P_i$  for every region  $G_i$  so that  $\sum_i P_i = 1$ , and so that the likelihood of the data  $L = \prod_{j=1}^m \sum_{i=1}^n P_i p_{ij}$  is maximized.*

As shown in [5] the likelihood objective function is concave, and the maximization of this function is polynomially solvable since there is a separation oracle as long as the  $p_{ij}$  coefficients are fixed. We present here an Expectation-Maximization (EM) algorithm for the MLEI problem. Since this problem is concave, the EM algorithm will converge to the optimal solution.

### 2.1 EM Algorithm for Inferring Expression Levels

We now describe an algorithm for solving the MLEI problem. We are searching for  $P = \{P_1, P_2, \dots, P_n\}$  such that the likelihood of the data is maximized. Let  $M$  be the underlying true unobserved matching of reads to regions. Then the following is an EM algorithm that searches for  $P$  that maximizes  $L(P; R)$ . Let  $P^{(t)}$  be the current estimate of  $P$ .

**E step:**

$$\begin{aligned} Q(P|P^{(t)}) &= E_{M|R, P^{(t)}}[\log L(P; R, M)] \\ &= E_{M|R, P^{(t)}}\left[\sum_{i=1}^m (\log P_{M(i)} + \log p_{iM(i)})\right] \\ &= \sum_{i=1}^m \sum_{j=1}^n [(\log P_j + \log p_{ij}) \times \frac{P_j^{(t)} p_{ij}}{\sum_{j=1}^n P_j^{(t)} p_{ij}}] \end{aligned}$$

**M step:**

$$\begin{aligned} P^{(t+1)} &= \arg \max_P Q(P|P^{(t)}) \\ &= \arg \max_P \left[ \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log P_j + \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log p_{ij} \right] \end{aligned}$$

where  $a_{ij} = \frac{P_j^{(t)} p_{ij}}{\sum_{j=1}^n P_j^{(t)} p_{ij}}$ . Given that  $p_{ij}$  (the probability of read  $j$  if it came from region  $j$ ) are fixed, maximizing the above function reduces to finding

$$P^{(t+1)} = \arg \max_P \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log P_j = \sum_{j=1}^n \left( \sum_{i=1}^m a_{ij} \right) \log P_j$$

It can be easily shown that the maximum is achieved at:

$$P_j^{(t+1)} = \frac{\sum_{i=1}^m a_{ij}}{\sum_{i=1}^m \sum_{j=1}^n a_{ij}}, \forall j$$

Since the likelihood function is concave [5], the above EM is guaranteed to converge to the optimal solution. Although it does not have the same polynomial time guarantee as the method in [5], in practice it outperforms the HAPLOFREQ method of [5] and provides a basic framework for the extension of the MLEI problem to the case of joint estimation of expression levels and variants where the sequenced sample differs from the reference genome. Since Single-Nucleotide Polymorphisms (SNPs) are the most common source of variation in the human genome we focus primarily on single nucleotide variants although other type of variants can be easily incorporated into the model. The model of reads with SNP variants is more realistic and may also be more powerful for certain cases since SNPs can be used to distinguish genomic locations in homologous regions. We demonstrate in the Results section that the solution obtained by the EM more accurately estimates the gene expression levels  $P$ , than the heuristic methods of either ignoring the multireads altogether or dividing them among the regions they map to.

### 2.2 Joint Estimation of Expression Levels and SNP Variants

In the above formulation we implicitly assumed that the probabilities  $p_{ij}$  were fixed and easy to compute since we had a fixed reference dataset. All differences between reads and reference were assumed to be due to errors and  $p_{ij}$  was simply a function of our model parameters. In practice however, the sequenced DNA may be slightly different than the reference genome, particularly in SNP positions. To model the SNP locations, we introduce a variable  $X_k = \{X_k^1, X_k^2\}$  with  $X_k^1, X_k^2 \in \{A, C, T, G\}$  for each genomic position  $k$ , which denotes the genotype of the sequenced sample at that location. The values of  $X_k$  are unknown and they have to be inferred. We can assume we have a prior distribution of  $X_k$  which corresponds to the distribution of the allele frequencies in the genome – this distribution can be empirically estimated (depending on the ancestry of the sample) from the HapMap[3] data, and particularly the ENCODE[2] regions, as well as the 1000 genomes project when the data becomes available. Particularly, we can have an estimate of the distribution of allele frequency across positions that are not known to be SNPs based on the ENCODE regions, and for the other positions we have their allele frequencies from dbSNP or from HapMap. Now, if the plausible alignment of read  $r_i$  to region  $G_j$  spans the positions  $X_1, \dots, X_l$ , assuming that sequencing errors are independent of each position, we can write  $p_{ij}$  as:

$$p_{ij} = \prod_k \gamma(X_k, r_i^k, k)$$

where,

$$\gamma(X_k, r_i^k, k) = \begin{cases} \epsilon(k), & \text{if } X_k^1 \neq r_i^k, X_k^2 \neq r_i^k \\ 1 - \epsilon(k), & \text{if } X_k^1 = r_i^k, X_k^2 = r_i^k \\ 0.5, & \text{otherwise} \end{cases}$$

$\epsilon(k)$  is the error rate function in a read at position  $k$ . The dependency of the error rate on the position comes from technological constraints as the error

rate is expected to increase with the length of the reads (see [4] for empirical estimates of Solexa error rates). Based on this, the problem of joint estimation of expression levels and SNP variants can be defined as follows:

**Definition 2 (MLEI-SNP).** *Given a set of reads  $r_1, \dots, r_m$  and a set of regions  $G_1, \dots, G_n$ , find a probability  $P_i$  for every region  $G_i$  and genotype  $X_k = \{X_k^1, X_k^2\} \in \{A, C, T, G\}^2$  for every location  $k$ , so that  $\sum_i P_i = 1$ , and so that the likelihood of the data  $L = \prod_{j=1}^m \sum_{i=1}^n P_i p_{ij}$  is maximized, where  $p_{ij} = \prod_{k=1}^l \gamma(X_k, r_i^k, k)$ .*

**EM extension with SNP variants.** In order to maximize the likelihood of the data, we are now looking for both  $P = \{P_1, P_2, \dots, P_n\}$  s.t.  $\sum P_i = 1$  and genotype calls  $X = \{x_1, \dots, x_k\}$  for every genomic location so that the likelihood of the data  $L(P, X; R) = \prod_{j=1}^m \sum_{i=1}^n P_i p_{ij}$  is maximized, where  $p_{ij}$  is defined as before:

$$p_{ij} = \prod_k \gamma(X_k, r_i^k, k)$$

The EM algorithm can be adapted as follows:

**E step:**

$$\begin{aligned} Q(P, X | P^{(t)}, X^{(t)}) &= E_{M|R, P^{(t)}, X^{(t)}} [\log L(P, X; R, M)] \\ &= E_{M|R, P^{(t)}, X^{(t)}} \left[ \sum_{i=1}^m \log P_{M(i)} p_{iM(i)} \right] \\ &= \sum_{i=1}^m \sum_{j=1}^n \left[ (\log P_j p_{ij}) \times \frac{P_j^{(t)} p_{ij}^{X^{(t)}}}{\sum_{j=1}^n P_j^{(t)} p_{ij}^{X^{(t)}}} \right] \end{aligned}$$

**M step:**

$$\begin{aligned} (P^{(t+1)}, X^{(t+1)}) &= \arg \max_{P, X} Q(P, X | P^{(t)}, X^{(t)}) \\ &= \arg \max_{P, X} \left[ \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log P_j p_{ij} \right] \\ &= \arg \max_{P, X} \left[ \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log P_j + \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log p_{ij} \right] \end{aligned}$$

Since the two terms in the above equation are independent we can maximize them separately. Just as before the first term in the equation above is maximized when  $P_j^{(t+1)} = \frac{\sum_{i=1}^m a_{ij}}{\sum_{i=1}^m \sum_{j=1}^n a_{ij}}$ , where  $a_{ij} = \frac{P_j^{(t)} p_{ij}^{X^{(t)}}}{\sum_{j=1}^n P_j^{(t)} p_{ij}^{X^{(t)}}}$ .

The second term is more complicated as we need to find  $X^*$  that maximizes  $\sum_{i=1}^m \sum_{j=1}^n a_{ij} \log p_{ij}$ . However, since the term depending on  $p_{ij}$  is a log of a

product, we can decompose it into independent contributions for each genomic location  $k$  and optimize each  $X_k$  independently. Namely,

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log p_{ij} &= \sum_{i=1}^m \sum_{j=1}^n a_{ij} \log \prod_k \gamma(X_k, r_i^k, k) \\ &= \sum_{i=1}^m \sum_{j=1}^n a_{ij} \sum_k \log \gamma(X_k, r_i^k, k) \\ &= \sum_k \sum_{\text{read } i \text{ spans } k} a_{ij} \log \gamma(X_k, r_i^k, k) \end{aligned}$$

and thus we set

$$X_k^{(t+1)} = \arg \max_{X_k=(x_k^1, x_k^2)} \sum_{\text{read } i \text{ spans } k} a_{ij} \log \gamma(X_k, r_i^k, k)$$

In practice we can speed up the computations by noticing that in the  $M$  step when finding new estimates for  $X_k^{t+1}$  we only need to consider locations  $k$  at which there are at least  $c > 0$  mismatches to the reference.

### 3 Results

In this section we present results on both simulated and real data sets showing the superior accuracy of our approach when compared to three previously proposed heuristic approaches for this problem. The first method we compare to is the standard method that ignores all multireads and estimates the expression levels  $P_i^{uniq}$  as the percentage of unique reads mapped to region  $i$  amongst all uniquely mapped reads. The second method estimates  $P_i$  by dividing the ambiguous reads uniformly between each region it maps to. Namely,  $P_i^{uniform} = \frac{1}{m} \sum_{j:j \text{ maps to } i} \frac{1}{h(i)}$ , where  $h(i)$  is the number of locations read  $r_i$  maps to. A more intuitive approach [6,15] is to divide each read amongst each location it maps to according to weights, where the weights are given by the distribution of the uniquely mapped reads in those regions; we denote this method as the *weighted* approach.

**Performance measures.** We use two correlated measures for the distance between the estimated and true distributions of the RNA expression levels  $P$ .  $P_i$  denotes the true expression level of a gene and  $\hat{P}_i$  is the estimated expression level. The first measure we use, the *error rate*, is computed as  $\frac{1}{n} \sum_i \frac{|P_i - \hat{P}_i|}{P_i}$  and it quantifies the average distance between the true and the estimated expression level in a region. A second approach to measure the accuracy of the estimates is the “goodness of fit” measure between the two distributions, in terms of *chi-square difference*:  $\sum_i \frac{(P_i - \hat{P}_i)^2}{P_i}$ . This measure is of particular interest as it is correlated to the power to detect differentially expressed regions.

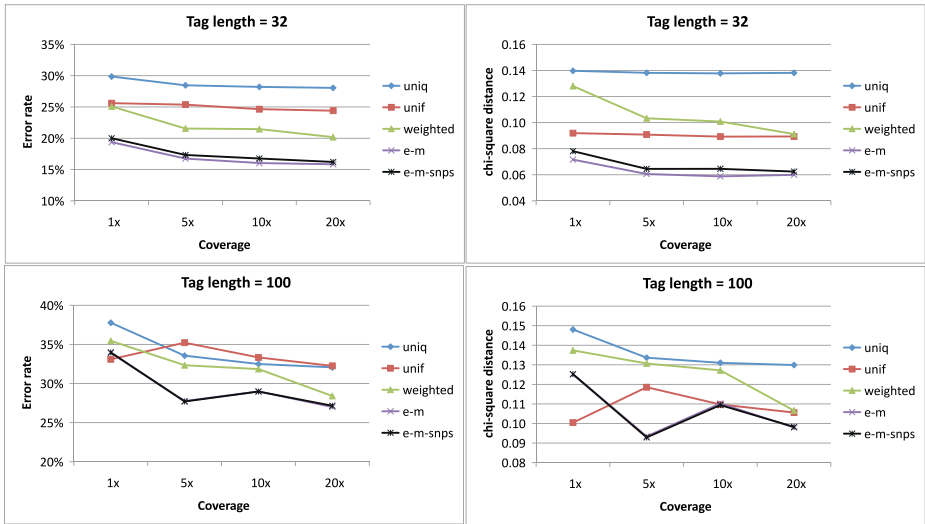
**Simulated Datasets.** In the first set of experiments we assessed the performance of our framework on RNA-seq by simulating short reads based on chromosome 1

from the human genome as a reference sequence. We focused on known homologous genes since they are the genes that are most affected by multireads. To do this, we downloaded the 756 human homologous genes from chromosome 1 from the Homologene[10] database. We removed all overlapping genes and genes with no other homologs in human resulting in 51 genes over 95kb.

The human reference genome does not contain information about possible polymorphisms, however it is expected that we will see both homozygous and heterozygous variants when sequencing a random individual in comparison to the reference. Given that the sequencing sample is different from the reference at a locus where the SNP allele frequency is  $f$ , the probability for a heterozygote is  $2f(1-f)$  and for a homozygous variant different from the reference is  $f^2(1-f) + f(1-f)^2 = f(1-f)$ . Thus, given that a site is different from the reference, the probability of a heterozygote is  $2/3$ , and of a homozygote is  $1/3$ , regardless of the allele frequency  $f$ . As done elsewhere[13], we used this observation when simulating a sample. First we pick a set of variants (where the sample differs from reference) with a rate of  $10^{-3}$  (which is the approximate frequency of SNPs in the genome) and then we randomly set  $2/3$  of the variants as heterozygous and  $1/3$  homozygous. In order to make the simulations as close to the actual data as possible, we also picked genotypes for the sample at known HapMap SNPs from the distribution given by the HapMap CEU frequencies.

For each of the 51 homologous genes we randomly chose  $P_i$  according to the uniform distribution, and normalized so that  $\sum_i P_i = 1$ ;  $P_i$  represents the true expression rate for gene  $i$ . We generated  $x_i$  reads for this region, where  $x_i = \frac{C \times L(i) \times P_i}{T}$ .  $C$  is a parameter of the simulation denoting the coverage rate,  $L(i)$  is the length of the gene in base-pairs (we only count the exons) and  $T$  is the length of the read. Although currently available NGS technologies such as Solexa[9] or ABI Solid[8] produce reads of length 20 to 40 base-pairs it is expected that the read length will increase dramatically to up to 100 bp and more in the near future. For this reason, we use simulations for two tag lengths ( $T = 32$  and  $T = 100$ ) thus simulating both currently available technologies and future technological developments. For each read at every location we inserted errors using a rate of  $\epsilon = 0.01$ ; similar results were obtained on simulations using an empirical error model that was estimated by Dohm et al.[4] (data not shown). The reads were mapped to chromosome 1 hg18 using the bwa[12] mapping algorithm with default parameters.

**Inferring expression levels in homologous genes.** In our first set of results we compared the EM algorithms with or without SNP variant calling to previously employed methods. Figure 1 shows that both EM algorithms outperform the other methods for both 32 and 100 bp length reads as well as for the different accuracy measures. Indeed for reads of length 32 the error rate decreases from approximately 30% for the *uniq* method that uses only the uniquely mapped reads to approximately 20% for both EM methods. The improvement, although still substantial, is more modest for reads of length 100, probably due to a smaller number of multireads as compared to reads of length 32.



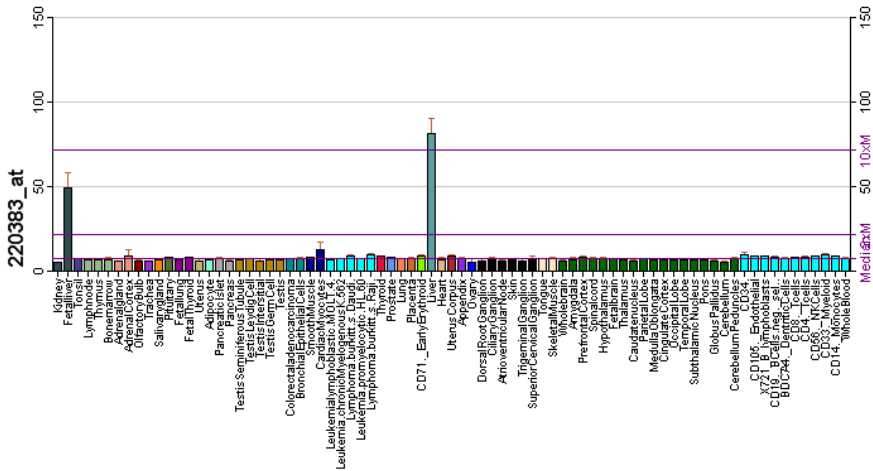
**Fig. 1.** Accuracy of gene expression inference based on simulated RNA-seq data for different read lengths and different accuracy measures. Results are given as averages over 100 simulated datasets. The EM methods outperform the heuristic methods of assigning reads as well as the approach of ignoring multireads.

To further highlight the effect of including the multireads in subsequent analyses as opposed to the general approach of using only the uniquely mapped reads we assessed the quality of SNP variant inference with or without multireads. To maintain a meaningful comparison, we called SNP variants based on unique reads under the same likelihood method for calling SNPs as in the EM algorithm of Section 2.2. Table 1 shows the true and false positive rates for SNP variant calling showing that the *e - m - snps* method outperforms the *uniq* method for all studied coverages when compared to the method that employs only the uniquely mapped reads.

**Detecting differential expression.** Using the same set of genes as before we simulated pairs of experiments with different expression levels for the genes. Using the true expression levels and a standard chi-square test ( $\alpha = 0.01$ ), we first computed a set of differentially expressed genes between the experiments which serve as the gold standard “true” differentially expressed genes. We assessed the capacity of identifying the differentially expressed genes when different methods were used for estimating  $P_i$ 's. The EM method shows the overall best performance, area under ROC curve of .83, compared to .75 for the *uniq* method and .81, .82 for the *unif* and *weighted* methods. For  $\alpha = 0.05$  cutoff, EM achieves (true positive, false positive) rates of (97.5%, 24.5%) compared to (88.4%, 20.8%) for *uniq* method, (95.9%, 26.6%) for *unif* and (96.6%, 26.4%) for *weighted* method.

**Table 1.** Variant calling rates on simulated datasets with reads of length 32 for various coverages. Results given in averages over 100 simulated datasets.

coverage	method	TPR	FPR
1x	uniq	18.00%	2.39E-05
	e-m-snps	18.26%	4.97E-05
5x	uniq	53.19%	3.27E-05
	e-m-snps	55.52%	3.99E-05
10x	uniq	69.67%	4.82E-05
	e-m-snps	73.55%	4.13E-05
20x	uniq	79.23%	3.50E-05
	e-m-snps	83.65%	2.26E-05

**Fig. 2.** Expression levels of gene ABCG5 in the GeneAtlas (<http://biogps.gnf.org>) project with high expression in Liver and Fetalliver. Gene ABCG5 is shown to be highly differentially expressed between Liver and Kidney in Marioni et al. [14] RNA-seq data only when using our EM method for inferring gene expression levels.

**Real dataset.** We also applied our methods to a real RNA-seq data set from Marioni et.al [14] consisting of two runs of an Illumina Genome Analyzer with half of the lanes containing human liver RNA and half kidney. We mapped all the reads with bwa [12] to the human genome sequence build hg18 and counted the number of reads in exons (we used the exon annotation of UCSC genome browser [7]). The read counts per gene were highly correlated across lanes and did not exhibit a lane effect for most lanes [14]. We used the data from lanes one and two from the first run to estimate kidney and liver expression levels. We used the *weighted* method and our EM method to estimate the read counts for each gene. In this case we do not know the true expression levels of the genes so we can not report which method is more accurate. Instead, we measure the



number of genes exceeding a  $5x \log_2$  fold change between each of the methods. For genes with uniquely mapped reads, these methods will perform identically, so we restricted our analysis to the 2207 genes with more than 200 multireads. For this set of homologous genes our EM method found 94 highly differentially expressed genes, while the *weighted* method reported only 86, a decrease of 8.5%. All of the genes found to be highly differentially expressed using the *weighted* method were contained in the set found using EM. To verify that the additional 8 genes we found using EM were not false positives we examined their expression levels in the GeneAtlas project [17], a comprehensive survey of gene expression in human tissues. For 7 of the 8 additional genes we found GeneAtlas expression levels were consistent with the EM findings; the probe intensities were greater than 50 in one tissue and less than 10 in the other. Figure 2 shows an example for the gene ENSG00000138075 (ABCG5). Note that ABCG5 has a known homolog ABCG8 so it is one of the cases that our method addresses. Only one of these eight genes predicted to be differentially expressed by EM, was not differentially expressed in the GeneAtlas. Overall, these data confirm the increased power of our method, suggesting that the additional differentially expressed genes found by the EM are true positives.

## 4 Discussion

Given the dropping cost of sequencing, and the numerous advantages RNA-seq has over expression array based experiments, it is likely that in the next future RNA-seq will become a pervasive choice for measuring cellular RNA expression levels. Many of the analyses conducted so far have utilized varying methods, and it is currently unclear which strategies will prove to be the most accurate and powerful. Considering the rich literature discussing proper analysis of microarray data over the last fifteen years, it is likely that methods for this new technology can be significantly improved.

This work addresses an important aspect of RNA-seq analysis; how to handle reads from homologous and repetitive elements that map to multiple genomic locations. Our results clearly show that naïve approaches significantly underestimate the true expression of homologous genes. Unlike previous heuristic approaches we present methods based on a rigorous probabilistic generative framework for an RNA-seq experiment and show that our approach consistently outperforms all previous attempts at solving this problem. We also applied our approach on a real RNA-seq data set to find several new highly differentially expressed genes when compared to previous approaches; these findings were confirmed by existing expression array data sets.

We have identified several areas of improvement that we plan to address in future work. Currently, our method is limited to the use of consensus genes and maybe improved by additionally modelling isoforms, splice variants, allelic heterogeneity, and un-annotated genes. In addition, the problem of multireads extends beyond RNA-seq experiments. For example, in both ChIP-seq and RIP-seq scenarios array based methods are replaced with an NGS approach and so

analysis methods must again handle multireads. Instead of determining the distribution of multireads as in RNA-seq, a binary signal is returned specifying whether or not a particular transcription factor binds to a specific genomic location. Solving the multiread problem in this context can potentially increase the power of detecting interesting loci, particularly when these loci fall within repetitive elements of the genome.

## References

1. Cokus, S.J., Feng, S., Zhang, X., Chen, Z., Merriman, B., Haudenschild, C.D., Pradhan, S., Nelson, S.F., Pellegrini, M., Jacobsen, S.E.: Shotgun bisulphite sequencing of the arabidopsis genome reveals dna methylation patterning. *Nature* 452(7184), 215–219 (2008) (03 2008/03/13/print)
2. The ENCODE Project Consortium. Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature* 447, 799–816 (2007)
3. The International HapMap Consortium. A second generation human haplotype map of over 3.1 million snps. *Nature* 449(7164), 851–861(2007) (10 2007/10/18/print)
4. Dohm, J.C., Lottaz, C., Borodina, T., Himmelbauer, H.: Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucl. Acids Res.* 36(16), e105 (2008)
5. Halperin, E., Hazan, E.: Haplofreq: Estimating haplotype frequencies efficiently. *Journal of Computational Biology* 13(2), 481–500 (2006) (PMID: 16597253)
6. Hashimoto, T., de Hoon, M.J.L., Grimmond, S.M., Daub, C.O., Hayashizaki, Y., Faulkner, G.J.: Probabilistic resolution of multi-mapping reads in massively parallel sequencing data using MuMRescueLite. *Bioinformatics* 25(19), 2613–2614 (2009)
7. <http://genome.ucsc.edu/>
8. <http://solid.appliedbiosystems.com/>
9. <http://www.illumina.com/pages.ilmn?ID=204>
10. <http://www.ncbi.nlm.nih.gov/homologene/>
11. Johnson, D.S., Mortazavi, A., Myers, R.M., Wold, B.: Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science* (2007) 1141319
12. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25(14), 1754–1760 (2009)
13. Li, H., Ruan, J., Durbin, R.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* 18(11), 1851–1858 (2008)
14. Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y.: RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research* 18(9), 1509–1517 (2008)
15. Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B.: Mapping and quantifying mammalian transcriptomes by rna-seq. *Nat. Meth.* 5(7), 621–628 (2008) (07 2008/07//print)
16. Schuster, S.C.: Next-generation sequencing transforms today’s biology. *Nat. Meth.* 5(1), 16–18 (2008) (01 2008/01//print)

17. Su, A.I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K.A., Block, D., Zhang, J., Soden, R., Hayakawa, M., Kreiman, G., Cooke, M.P., Walker, J.R., Hogenesch, J.B.: A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America* 101(16), 6062–6067 (2004)
18. Wang, Z., Gerstein, M., Snyder, M.: Rna-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10(1), 57–63 (2009) (01 2009/01//print)

# Cactus Graphs for Genome Comparisons

Benedict Paten<sup>1</sup>, Mark Diekhans<sup>1</sup>, Dent Earl<sup>1</sup>, John St. John<sup>1</sup>, Jian Ma<sup>2</sup>,  
Bernard Suh<sup>1</sup>, and David Haussler<sup>1</sup>

<sup>1</sup> Center for Biomolecular Science and Engineering,  
University of California Santa Cruz, CA, USA

<sup>2</sup> Department of Bioengineering,  
University of Illinois at Urbana-Champaign, Urbana, IL, USA

**Abstract.** We introduce a data structure, analysis and visualization scheme called a cactus graph for comparing sets of related genomes. Cactus graphs capture some of the advantages of de Bruijn and break-point graphs in one unified framework. They naturally decompose the common substructures in a set of related genomes into a hierarchy of chains that can be visualized as multiple alignments and nets that can be visualized in circular genome plots.

## 1 Introduction

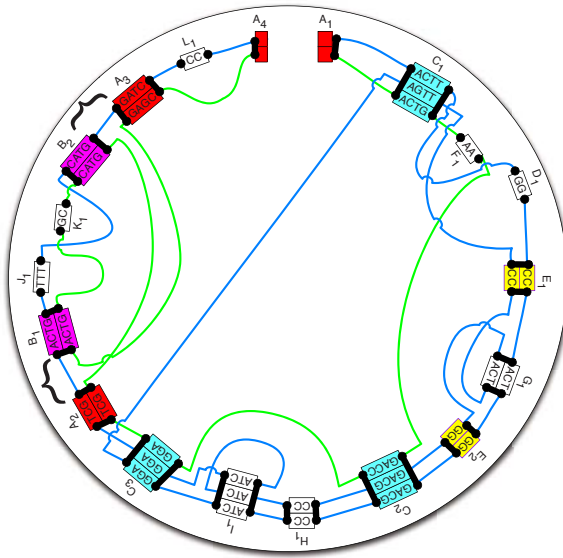
Genomes are often compared at a fine scale using multiple alignments [1] [2], which capture base-level differences, and at a large scale using circular genome plots [3] [4], which capture rearrangements. However, the changes between genomes exhibit structure at many intermediate levels as well. This structure is inherently nested, with small inversions inside of larger inversions, duplications within duplications, etc. We introduce a data structure called a cactus graph that captures the nested structure of genome comparisons. Applications of cactus graphs include the comparison of reference genomes from related species, comparison of structural variation between genomes of the same or different individuals within a species [5], and comparison of different somatic variants of an individual's germline genome, e.g. in cancer genomics research [6] [7].

The first step in genome comparison is to identify and align *segments* of DNA that are homologous between and within the genomes being compared. These segments may be, for example, coding exons, recognizably conserved noncoding elements, or large orthologous chromosomal regions of closely related genomes. A multiple alignment of a set of homologous segments is called a *block*.

Identification of the segments leaves behind stretches of unaligned DNA we call *adjacencies* between segments and at the ends of chromosomes. To make the two adjacencies at the opposite ends of a chromosome into proper adjacencies, we add a *cap* at each end representing the telomeres, and connect these two caps by adjacencies to the first and last segments. More generally, a cap can be the end of any sequence of DNA. Thus, when applying these conventions to represent an internal part of a chromosome, the caps are the ends of the segments flanking this internal part of the chromosome. We define a *thread* as a

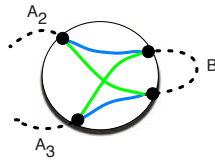
path of alternating adjacency and segment edges that is flanked by adjacencies connected to caps. The potential nesting of threads makes it easier to represent hierarchical structures within the chromosomes of the genomes being compared.

Caps naturally inherit the homologies of the segment ends that define them. Additional homologies can be defined *a priori* for chromosome telomeres, so that all caps, be they internal segment ends or chromosome telomeres, are treated in the same fashion. A family of homologous caps is called an *end*. Figure 1 shows an example of two different threads traversing a set of ends, blocks and adjacencies.

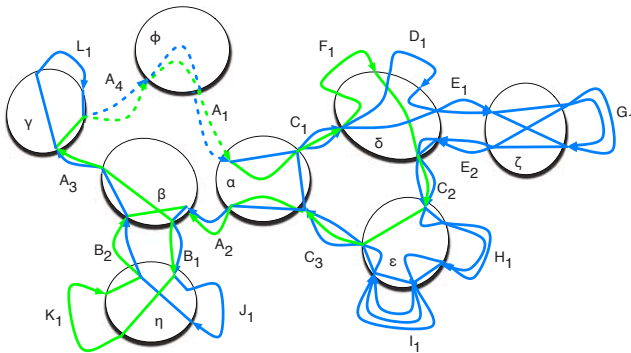


**Fig. 1.** A circular genome style plot showing a complete net with examples of chains and threads. Blue and green lines depict two homologous threads traversing a series of segments in blocks and the joining adjacencies. All aligned boxes are blocks except  $A_1$  and  $A_4$ , which are ends representing telomeres. The ends of the blocks and the ends of the telomeres are mapped as filled black rectangles on the edges of the aligned boxes. These are the nodes in the complete net. DNA bases within the adjacencies are not shown. Ignoring the unseen sequence within the adjacencies and starting at  $A_1$ , the blue thread gives the sequence ACTTGGCCACTGGGACGCCATC-GGAAGTTCCagtGGGACGCCATCATCGGATCGACTGTTTTCATGGATCCC. The green thread gives the sequence ACTGAAGACCGGATCGcatggccagtGAGC. The lower case “agt” in the blue thread represents the reverse complement of the bottom segment of block  $G_1$ , which is traversed right-to-left in the blue thread and similarly, the lower case segment in the green thread is the reverse complement of segments in  $B_2$ ,  $K_1$  and  $B_1$ , also traversed right-to-left. Chains containing more than one block/end are given distinct colors; for example chain A has two blocks,  $A_2$  and  $A_3$ , and two ends,  $A_1$  and  $A_4$  in it, all colored red. The large curly brackets highlight the four ends of the subnet shown in Figure 2.

A *net* is a graph in which each node is an end and each edge represents a set of adjacencies between the caps in the two ends it connects. A *complete net* for the comparison of a set of genomes has a node for each end of every block and a node for each telomere end. There is an edge between two nodes whenever there is an adjacency between them in any of the genomes being compared. Usually the nodes are laid out on a circle and the edges are geodesics that cross the circle (Figure 1) [3] [4]. Complete nets quickly get very dense and hard to interpret with growing genome size and genome distance, thankfully they can often be decomposed into smaller components. The cactus graph provides an organizing principle in which simpler subnets and nested substructures can be extracted from complete nets. For example, the four ends highlighted by curly braces in Figure 1 form a connected component of adjacencies for which we can construct a net as shown in Figure 2. In the “blue” genome, they appear as  $A_2 B A_3$



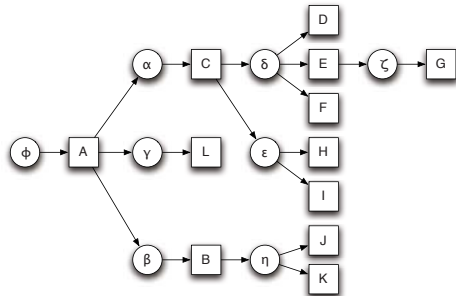
**Fig. 2.** A net for the ends highlighted by curly brackets in Figure 1. The net is composed of four ends: an end of  $A_2$ , and end of  $A_3$ , and both ends of the chain  $B$ , which is composed of  $B_1$  and  $B_2$ . The ends are represented by the four filled circles on the larger circle. The adjacencies between the ends of the elements are the colored green and blue lines, the coloring indicating the two respective threads of Figure 1.



**Fig. 3.** A cactus graph with embedded net substructures for the chains and threads in Figure 1. The blue and green lines again depict the two threads. Each net node is shown in a circle; the origin net is  $\phi$ . Block and end edges are depicted with multiple arrows, representing the different threads traversing them. The dotted arrows of  $A_1$  and  $A_4$  indicate they are inherited ends. The lines within the circles represent the adjacencies. The dotted lines in net  $\phi$  represent the backdoor adjacencies connecting the dead ends of  $A_1$  and  $A_4$ .

and in the “green” genome they appear as  $A_2 -B A_3$ , where the negative sign denotes reverse complement. This inversion is represented cleanly in a simple subnet, separable from the larger complete net.

Each node of the cactus graph is a subnet of a complete net, as determined by the construction in Section 2.3 below, and each edge is a block. Every block in the genomes being compared appears as an edge, and every adjacency between segments or from a segment to a telomere cap is represented in one of the subnets. The cactus graph consists of a single connected component that is composed of a set of simple cycles, i.e. cycles such that no node is used twice, such that any two simple cycles intersect at at most one node (Figure 3). This property gives it its “cactus-like” appearance. Each simple cycle in the graph has an orientation that determines the direction of each edge on the cycle (see Section 2.5). All the telomere ends are contained in a single subnet represented by a node called the *origin*. A hierarchical set of *chains* is defined as follows. For each simple cycle that includes the origin we define a *child chain* by concatenating the blocks represented by the edges of the cycle in the order that they appear, starting from the first outgoing edge from the origin. Each node along this cycle, apart from the origin, represents a *link* in the chain. Conceptually, the link consists of that node and the entire sub-cactus graph that is attached to the chain at that node, i.e. the smaller cactus you would get if you pruned off this piece and replanted it. The origin node is called the *parent* of the child chain. Traversing outwardly from child chains of the origin node the definition of further chains proceeds recursively. Each node in one of the previously found simple cycles for which we have not yet defined a chain set becomes a new origin-like node, and we define child chains for it in the manner above, until all nodes have been explored and all chains are children with unique parents. This recursion results in a hierarchical structure called the *cactus tree*, a bi-layered tree consisting of parent subnets describing the relative order and orientation of their child chains, and these chains in turn containing a subnet in each of their links that describes further chains nested inside these links, and so on (Figure 4). This hierarchy represents the organization of the substructures shared between the genomes at various levels, from large chromosomal regions down to individual bases.



**Fig. 4.** A cactus tree for the cactus in Figure 3. Nets are shown as circles, chains as squares. The tree is bi-layered, with alternating net and chain layers.

In this manner, a cactus graph partitions a set of genomes into nested structures represented by chains and subnets, which can be visualized using alignments and circular plots, respectively. Cactus-graph-derived chains and subnets are analogous to those introduced in [8], but not identical. The theory behind the cactus graph [9] generalizes the notion of components and their hierarchies defined by Bergeron et al. for the study of rearrangements between pairs of genomes [10] [11]. The abstract combinatorial notion of a cactus graph, discussed further below, has also been used in many different optimization problems, including graph decomposition [12], optimal traffic [13] and facility location problems [14], and electrical circuits [15].

## 2 Results

### 2.1 Basepairs, Chromosomes and Genomes

We start by linearizing all the circular chromosomes in a set of input genomes, by breaking each of them at an arbitrary point. Let  $S$  be the resulting set of linear *input chromosomes*. Here we assume the input chromosomes are single complete sequences, in Section 1.1 of the appendix we consider the following construction stages with missing data. Mathematically a chromosome is just a circular string of signed symbols taken from a fixed alphabet of possible symbols. To avoid being excessively abstract, we will assume that the alphabet is just the symbols  $\{A/T, T/A, C/G$  and  $G/C\}$  for basepairs, with the understanding that the usual rules for reverse-complementing basepairs apply.

### 2.2 Homology

We say that two basepairs in  $S$  are *homologous*, denoted  $x \sim y$ , if they are related to each other by some given biological definition of relatedness, e.g. if they descend from a common ancestral basepair that existed a certain time in the past. For the purposes of this paper we require only that the notion of homology between basepairs be an equivalence relation. Two strings  $x = x_1 \dots x_n$  and  $y = y_1 \dots y_n$  are homologous if their bases are homologous, i.e.  $x \sim y$  if  $x_1 \sim y_1$ ,  $x_2 \sim y_2$ , ..., and  $x_n \sim y_n$ .

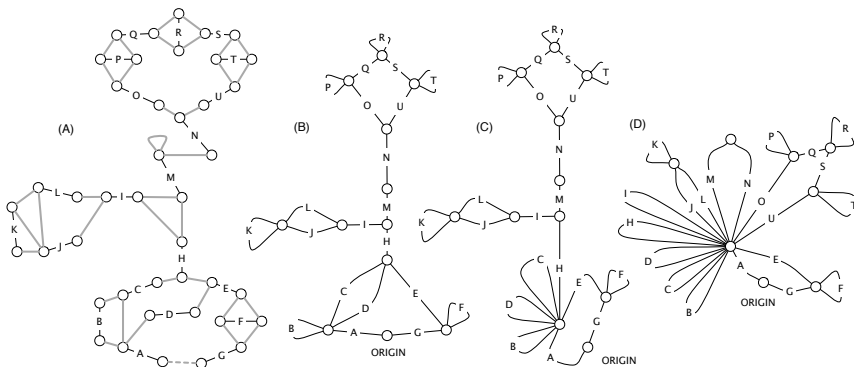
### 2.3 Blocks, Ends, Adjacency Graphs and Cactus Graphs

The goal is to represent the common structure between substrings of homologous bases in  $S$ . To do this we model two types of aforementioned homology structure, blocks and ends. A block is formally a maximal set of maximal-length homologous strings, represented by a gapless alignment of these strings. The blocks are shown as boxes containing gapless alignments in Figure 1. Blocks are defined by first forming a column consisting of all of the basepairs homologous to the given basepair, and then adding further columns to the left and right whenever the adjacent bases are all also homologous to each other. The horizontal rows



of the block, the previously defined segments, are the strings representing the homologous sequences of DNA aligned in the block. Ends are maximal sets of homologous caps. We define two types of end for our graphs, *block ends*, which are the ends of blocks and *inherited ends*, which include the previously mentioned telomeres, and which will also include block ends from higher level problems when we define multi-level cactus graphs (Section 2.6).

The *adjacency graph* (Figure 5 (A)) is a graph with a node for every end and an *adjacency edge* between two ends if there is an adjacency, potentially containing a nonempty substring from the input sequence, between a cap in one of the ends and a cap in the other end, i.e. if the caps would abut except for a possibly non-empty intervening adjacency substring in the input chromosome in which they appear. Self-edges are allowed in the adjacency graph, and occur when two homologous caps in opposite orientation share an adjacency. Multi-edges are not included in the adjacency graph, i.e. there is at most one adjacency edge between any two nodes, even if there are several adjacencies between them; in this case the adjacency edge is labelled with the set of adjacencies and their substrings, which uniquely pair caps between the ends they link. Unlike blocks the substrings within the adjacencies of an adjacency edge are not assumed to be homologous and are therefore not aligned. The two ends of each block are also connected by an edge in the adjacency graph; these edges are called *block edges*, and are labelled with the oriented set of aligned segments of the block they represent. In addition to the block edges the adjacency graph also includes *end edges*; for each inherited end the adjacency graph includes one end edge that connects the node representing the inherited end to a special *dead-end* node. All dead end nodes are in turn connected in a clique by unlabeled *backdoor adjacency*



**Fig. 5.** Examples graphs in the stages of construction of the final cactus graph from an initial adjacency graph. (A)  $G_0$ , an example adjacency graph. All black edges represent blocks except the black edges  $A$  and  $G$ , which are end edges; adjacency edges are grey, a backdoor adjacency (dotted grey edge) attaches the dead end nodes to one another. (B)  $G_1$ , the same graph after the collapse of the adjacency components. (C)  $G_2$ , after the collapse of the 3-edge connected components. (D)  $G_3$ , after modifications to bridge edge components to make the graph Eulerian.

edges. The adjacency graph is almost equivalent to a multi-breakpoint graph [16], and is related to various types of de Bruijn graphs used in comparative genomics and sequence assembly [17] [18].

Let  $G_0$  be the adjacency graph. The cactus graph is built from  $G_0$  in a series of steps, as illustrated in Figure 5.

(1) Ignoring the block and end edges, we compute the connected components of  $G_0$  formed by the adjacency edges only. These are called *adjacency-connected components*. All dead ends will be in a single component which we call the *origin component*. The graph  $G_1$  represents this decomposition of  $G_0$  into the resulting adjacency-connected components (Figure 5(B)). There is a node in  $G_1$  for every adjacency-connected component in  $G_0$ . The graph  $G_1$  has only block and end edges, no adjacency edges. Two nodes  $X$  and  $Y$  in  $G_1$ , representing (not necessarily distinct) adjacency-connected components in  $G_0$ , are connected by an edge in  $G_1$  for every block or end edge in  $G_0$  from some  $x \in X$  to some  $y \in Y$ . Thus, the graph  $G_1$  is formed by *merging* adjacency-connected nodes in  $G_0$  and retaining only the block and end edges in the merged graph. We call the node in  $G_1$  and subsequent graphs containing the origin component of  $G_0$  the *origin node*.

(2) We compute the decomposition of  $G_1$  into 3-edge connected components using the linear time algorithm in [19]. To define this decomposition, we say that two nodes  $x$  and  $y$  in  $G_1$  are equivalent if there is no set of up to two edges in  $G_1$  which, upon removal, disconnect  $G_1$  in such a way that there is no path from  $x$  to  $y$ . Thus, two nodes are equivalent if it takes the removal of 3 or more edges to disconnect them. The equivalence classes of nodes are called *3-edge connected components*. The graph  $G_2$  represents this decomposition (Figure 5(C)). It has one node for each 3-edge connected component. Two nodes  $X$  and  $Y$  in  $G_2$  are connected by an edge for every edge in  $G_1$  between some node  $x \in X$  and some node  $y \in Y$ . Thus, the graph  $G_2$  is formed by merging equivalent nodes in  $G_1$ . The theory of graph decomposition into 3-edge connected components shows that  $G_2$  is in fact a cactus graph in the combinatorial sense. However, it is not yet *the* cactus graph.

(3) Finally, to construct the cactus graph, we *fold in* the tree-like structures in  $G_2$  to obtain an *Eulerian cactus graph*  $G = G_3$  (Figure 5(D)). Formally, an edge in  $G_2$ , or indeed in any graph, is called a *bridge* if its removal disconnects the connected component in which it is contained. Consider the subgraph formed by only the bridge edges. It is easy to see that this subgraph is a *forest*, i.e. a collection of disjoint trees. In the fold-in process, for each such tree, we merge all leaf nodes and branching nodes into a single *tree loop node*. Only the non-branching internal nodes in the tree are left out of this merge, and appear on simple cycles emanating from the tree loop node, along with other cycles that were already present before this merge step. It is easy to see that the resulting graph  $G$  is also a cactus graph with one origin node. In fact, every node is either in a unique simple cycle or is the unique intersection of two or more simple cycles. Thus, all the nodes in  $G$  have an even number of edges incident upon them, i.e. are of *even degree*. We refer to a graph with even degree nodes as

an *Eulerian graph* after Euler's famous "Seven Bridges of Königsburg" example demonstrating that every connected component in such a graph must have a path through it that uses every edge exactly once and returns to its point of origin, a so-called *Eulerian circuit*.

Each node in the cactus graph  $G$  represents a set of block ends. The caps from these block ends are connected by a net structure as defined above, in which two caps are connected by an adjacency in which they appear. With the exception of the origin node, the net for a node defines a *perfect matching* between the caps incident upon the node, i.e. a pairing that includes each cap exactly once. The net for the origin node contains the set of dead ends, connected in a clique, and a set of non-dead ends which by definition must be connected to one another in a perfect matching. To construct a perfect matching for the origin node the backdoor adjacencies connecting the dead end nodes are removed and, using the fact that there are an even number of dead ends, they are replaced with a perfect matching. For any circular chromosomes we match their two dead ends by a backdoor adjacency to ensure that a thread which traverses them contains the adjacency which was originally broken when the circle was linearized. Otherwise the matching is arbitrary.

All adjacencies that occur between caps in the input sequences are represented in the nets of  $G$ . Every adjacency is represented in the net for the node to which it maps via the construction above. Thus, after we construct the perfect matching for the dead end nodes, when we trace the connected threads through the graph  $G$ , we recover precisely the set  $S$  of input chromosomes and a set of backdoor adjacencies, one backdoor adjacency being present each time we traverse between two chromosome ends. The perfect matching constructed between the dead ends defines the order in which threads traverse the input chromosomes. In this sense,  $G$  is a structured representation of  $S$ . To formalize this representation, the net structure for caps incident on each node and the segments of each block represented by an edge are both considered to be part of the cactus graph  $G$ , as node and edge substructures, respectively.

## 2.4 Traversals and Fundamental Cycles

A path in a graph is a sequence of edges  $(n_0, n_1), (n_1, n_2), (n_2, n_3), \dots, (n_{k-1}, n_k)$  that share intermediate nodes  $n_1, \dots, n_{k-1}$ . It is *simple* if it does not use the same intermediate node twice. It is a *cycle* if the first and last nodes are identical. A *traversal* of a simple cycle  $c$  is a path that uses only edges from  $c$ , with direction of travel on the edge indicated by sign. For example, if  $c$  is the simple cycle composed of edges 1 2 3 4 5, then  $t = 2\ 3\ 4\ -4\ 4\ 5\ 1\ 2\ 3\ -3\ -2\ -1\ 1\ 2\ -2$  is a traversal of  $c$ . In general, a traversal starts and ends at arbitrary nodes in the simple cycle, and each symbol in the traversal represents a move forward or backwards in it.

A simple cycle  $c$  in a graph  $G$  is *fundamental* if for any path  $p$  in  $G$ , if we ignore all edges in  $p$  that are not in the cycle  $c$ , we obtain a traversal of  $c$ . It is easy to see that  $c$  is fundamental if and only if there are no edges between nodes of  $c$  other than the edges of  $c$  itself and for every node  $n$  on  $c$  that is also connected

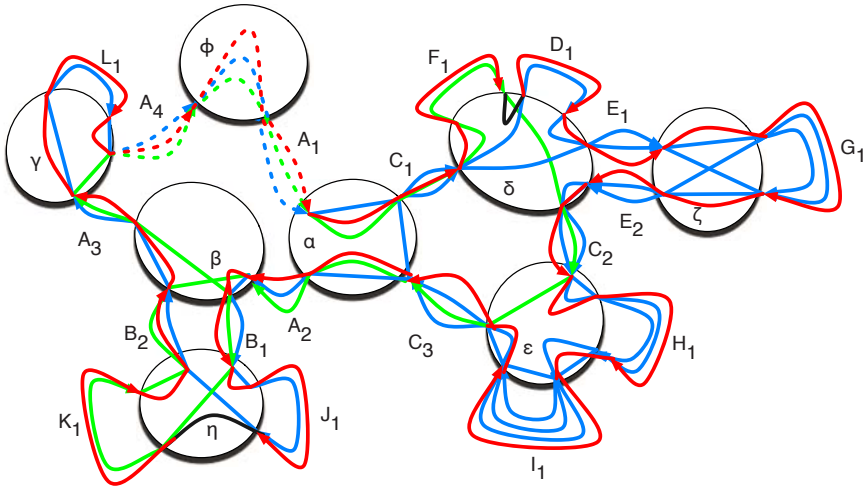
to a node not on  $c$ , removal of  $n$  disconnects the graph. When  $c$  is a fundamental cycle in  $G$ , it captures an invariant substructure within all paths in  $G$ . A cactus graph has the special property that all its simple cycles are fundamental. This follows directly from the fact that any two simple cycles intersect at at most one node. Let  $G$  be the cactus graph constructed as described above from a set of input chromosomes  $S$  and let  $c$  be a simple cycle in  $G$ . Let  $S_c$  be the set of chromosomes obtained from  $S$  by ignoring all the basepairs from blocks that are not in  $c$ . Then because  $c$  is fundamental, it follows that every chromosome in  $S_c$  is a traversal of  $c$ . Thus, the simple cycles in  $G$  represent universal substructures of the chromosomes of  $S$ .

## 2.5 Canonical Cactus Graph Layout and Reference Genomes

We define a cyclic *reference ordering* for the block ends (i.e. edges) incident on each node in the cactus graph in such a way that the two ends of each simple cycle incident on the node are adjacent in this ordering. Adjacencies in this cyclic order between ends of the same cycle are called *cycle identity adjacencies* and adjacencies between distinct cycle ends are called *cycle change adjacencies*. The cyclic order alternates between identity and change adjacencies. A planar embedding of the cactus graph that satisfies the cyclic order for edge incidences on each node is called a *canonical layout*. The key property of the cactus graph, that two simple cycles intersect at at most one node, implies that there is an underlying tree structure to the simple cycles, as we have already seen reflected in the cactus tree representing the hierarchy of chains, which guarantees that there always exists a planar (i.e. non-edge crossing) canonical layout of the cactus graph.

For each edge in  $G$  we define a *reference sequence*, which may, for example, be a consensus sequence for the alignment of the block represented by the edge, or an inferred ancestral sequence for the block. We use the cycle change adjacencies to define the thread end connections between these reference sequences. For each node, these adjacencies form a perfect matching of the edges incident upon the node that we call the *reference edge matching*. The *reference genome* is obtained by traversing the edges of  $G$  in the Eulerian circuit defined by the reference edge matchings (i.e. cycle change adjacencies) in the nodes and concatenating the corresponding reference sequences (Figure 6). In particular, whenever we enter a node via the end of a block, we leave the node again via the block end that is adjacent to it in the cyclic ordering for that node, which is the unique end matched to it in the reference matching. This is like the threading through  $G$  for the input chromosomes, except in the reference genome there is exactly one thread traversing each block, representing the reference sequence for the block. It is easily verified that this results in a Eulerian circuit with a single circular chromosome suitable for layout in a circular net display.

The reference genome represents more of an organizing principle than an actual biological entity. It has the key property that each homology block appears in it exactly once. Thus, it serves as a universal reference coordinate space to which we can unambiguously map strings from all of the input chromosomes



**Fig. 6.** A reference genome for the cactus in Figure 3. The added red thread represents a traversal of every block in the cacti exactly once. Black lines represent added adjacency edges, not observed in any input genome, but necessary to complete the tour. For example, the indels represented by J and K in  $\beta$  are both included by the addition of an extra adjacency edge. The reference genome defined gives the ordering of the blocks and ends in the circular genome plot Figure 11.

in  $S$ . If instead we try to choose one of the input sequences in  $S$  itself as the reference genome, we run into the problem that duplications that are specific to that sequence make it ambiguous to map homologous copies from the other input sequences. Related to the problem of defining a reference genome, in Section 1.2 of the appendix we consider ambiguity and multiple possible tours of the cactus graph.

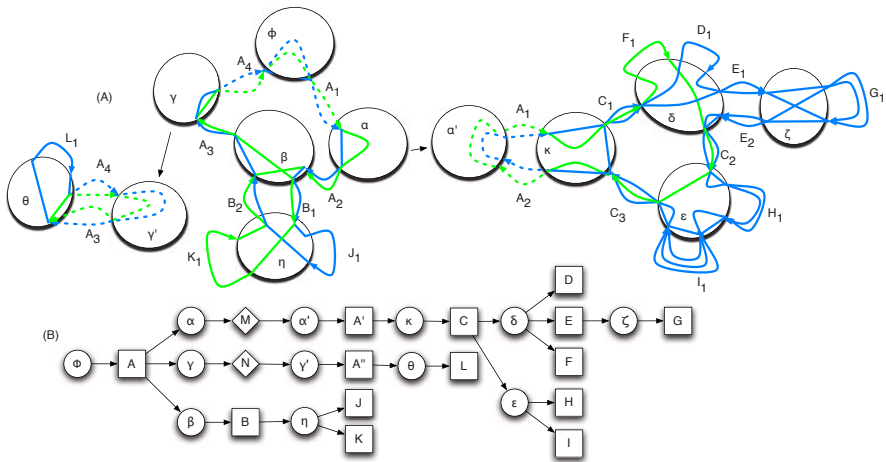
### 2.6 Multi-level Cactus Graphs

We do not insist that every basepair in the set of input genomes be contained within a block, but instead allow for bases to be contained within intervening unaligned adjacencies. This allows us to define “sparse” cactus graphs in which only a portion of the genomes are aligned, for example, one might initially define a high level sparse cactus graph in which the blocks were composed of homologous sets of exons. All bases outside the exons are contained in the adjacencies. Each node in the cactus graph is a net that is built from some set of these adjacencies. Now suppose we extend our notion of homology by aligning some of the bases that occur inside the adjacency edges in the nets. We can therefore define an orthogonal notion of recursion to that of the hierarchical structure of the cactus graph.

It is easier to define a high-level cactus graph using the segments and adjacencies at the lower level. Essentially, an adjacency at the higher level is a thread at the lower level. Formally, let us say that two threads are *similar* if they have

homologous caps at at least one end. A *group* is a minimal set of disjoint threads that is closed under similarity, i.e. a (pairwise) non-overlapping set of threads such that there are no threads that are similar to any of those in the set that are not already in the set, and there is no proper subset of these threads that has this closure property. A group is *self-contained* if there is no homology between any segment in a thread in the group with any segment outside of the threads in the group. Each net in a high-level cactus graph is a union of self-contained groups from the lower level segments and adjacencies.

There are two kinds of groups. A *link* is a group in which all the caps are part of two homology classes, i.e. two ends. A *tangle* is a group in which the caps form more than two homology classes (ends). We call a net whose adjacencies contain non-empty substrings of  $S$  *non-terminal* and conversely a net whose adjacencies contain only empty substrings *terminal*. In a multi-level cactus graph, for each self-contained group in a net, termed a *net contained group* (either link or tangle), we construct a *child cactus graph* in which: (1) Threads connecting caps from the group's ends are treated as linear chromosomes. (2) The group's ends become inherited ends in the child cactus graph, and thus map the boundaries between the *parent net* and the child cactus graph. (3) Homologies between segments within the threads form the blocks. Thus the adjacencies in the parent net are divided up into lower level segments and further adjacencies in the child cactus graph, repeating the process recursively that was used to construct the cactus graph containing the parent net. The recursion creating child cactus graphs can be continued until all non-terminal nets have defined child chains, and child



**Fig. 7.** Multi-level cactus graphs with embedded net substructure. (A) Middle: the cactus graph containing origin net  $\phi$  is the top level cactus. On the left and right (indicated by arrows) are child cacti extensions of the links in  $\gamma$  and  $\alpha$ , respectively. (B) The corresponding multi-layered cactus tree for the graphs shown in A. The diamond net contained group nodes represent the connection between the links in  $\alpha$  and  $\gamma$  and the child cactus graphs.

cactus graphs and thus all bases in  $S$  become part of a block in one cactus graph of the set of cactus graphs that comprise a *multi-level cactus graph* (Figure 7(A)). Just as the parent-child organization of chains and nets in a cactus graph can be represented by a bi-layered cactus tree, a multi-level cactus graph's chains, nets and net contained groups can be represented in a bi-layered multi-level cactus tree, Figure 7(B). The chain layer of a cactus trees in the multi-level cactus tree contains both chain and net contained group nodes, we consequently call this the *grouping layer* because it groups the ends in the nets above into child nets in the layer below, unless the node is a chain with no children, in which case the chain node is a leaf. The edges that emanate from a chain node in the cactus tree correspond to the links in the chain. For consistency, we say that link groups within chains are *chain contained links*. Conversely, net contained groups can be divided into *net contained links* and *net contained tangles*.

## 2.7 Implementation

We have implemented a three stage multi-level cactus graph recursion as outlined above. In the first stage of the recursion, after computing the pairwise alignments and forming the resulting homology classes which define the blocks, the subsequent construction stages as in Section 2.3 are followed. In the second and third stages of the recursion non-terminal nets are further, independently recursed upon to further “fill in” homologies in the sparser higher level cactus graphs. Within each non-terminal net the grouping of the block ends into net contained groups is made by the adjacency connectivity (as described) and further merging of these adjacency connected groups is made if the adjacencies they contain share homology according to the lower level alignment procedure. At the end of the three stages all non-terminal nets have defined net contained groups with attached lower level cactus graphs, such that the leaf nets (those with only chains as descendants) in the resulting multi-level cactus tree are always terminal.

In the first two stages pairwise alignments between sequences are computed using the LASTZ program ([http://www.bx.psu.edu/miller\\_lab/dist/README.lastz-1.01.50/README.lastz-1.01.50.html](http://www.bx.psu.edu/miller_lab/dist/README.lastz-1.01.50/README.lastz-1.01.50.html)). In the first stage LASTZ is run using the strict parameters: `-notransition -step=20 -nogapped`. In the middle stage the default LASTZ parameters are used. In the final stage a pairwise-HMM similar to the one in [20] is used to align each pair of sequences in the grouping in both the forward-forward and forward-reverse strand orientations, and those pairs of bases for which the posterior probability of alignment is greater than a threshold  $P$  (default  $P \geq 0.7$ ) are included in the set to be aligned.

In all stages spurious pairwise alignments, because the alignment relation is transitive, cause the over collapsing of the graph into blocks with a very large resulting number of segments; we term the number of segments in a block the block's *degree*. To overcome this problem we implement a recursive series of heuristics, to be fully described in forthcoming extended analysis, to repeatedly



**Table 1.** Statistics on the nets of the cactus trees. Region: region name. Bp size: total number of basepairs in the input sequences. T. nets: Total nets in the cactus tree, either 'all', including all nets in tree, 'nets', including only net contained groups or 'chains', including only chain contained groups.. Note the sum of net and chain contained groups is equal to all minus one (for the root node). Norm. relative entropy: Let  $N$  be a net in the set of all nets  $T$  in a cactus tree  $X$ . Furthermore let  $N_0 \dots N_{n-1}, N_n$  denote the ancestral path of nodes from the root net  $N_0$  of the cactus tree to the net  $N_n$ . Let  $P(X) = \sum_{N_n \in T} |b(N_n)|(\log_2(|b(N_n)|) + \sum_{i=0}^{n-1} \log_2(|c(N_i)|))$  and  $Q(X) = Z \log_2(Z)$ , where  $Z$  is the total number of basepairs in the input sequences,  $b(N)$  is the set of basepairs contained in blocks of the net  $N$ ,  $|b(N)|$  is the size of  $b(N)$ ,  $c(N)$  is the set of child nets (direct descendants) of  $N$  and  $|c(N)|$  is the size of  $c(N)$ . The total relative entropy is  $P(X) - Q(X)$  and the normalised relative entropy (NRE) is  $(P(X) - Q(X))/Z$ . The measure therefore reflects the balance of the tree. Children: The children of a net are its direct descendants nets in the subsequent net layer of the (multi layered) cactus tree. Results given for non-leaf nets only. Depth: The depth of a net is the number of nodes (excluding itself) on the path from it to the root node. Results for leaf nets only. (A leaf net is net with only chain descendants in the multi-layered cactus tree).

Nets														
Region	Bp Size	T. Nets			Relative Entropy			Children			Depth			
		All	Nets	Chains	P(X)/Z	Q(X)/Z	NRE	Max	Avg.	Med.	Min	Max	Avg.	Med.
ENm001	12993002	323029	105769	217259	23.63	35.44	11.81	1097	2.34	1	3	75	7.90	8
ENm002	7112290	164130	58132	105997	22.76	28.03	5.27	1179	2.25	1	3	19	6.63	7
ENm003	3538075	83837	28830	55006	21.75	29.66	7.90	710	2.30	1	4	35	6.86	7
ENm004	22314965	252560	89689	162870	24.41	38.67	14.26	2530	2.27	1	4	18	7.85	8
ENm005	11296788	276484	96670	179813	23.43	32.50	9.07	2832	2.29	1	3	45	7.66	7

**Table 2.** Statistics on the chains of the cactus trees. Region: region name. Type: categories of chains, either 'all', which includes all chains or '>= 2 B.', which includes only chains containing a minimum of two blocks. Total: total number of chains in the cactus tree. Per Net: numbers of child chains in each net. Link Number: number of links in chain. Block Bp length: number of basepairs in blocks of chain. Instance length: average number of basepairs in an instance of the chain, including both its blocks and intervening links.

Chains														
Region	Type	Total	Per Net			Link Number			Block Bp length			Instance Length		
			Max	Avg.	Med.	Max	Avg.	Med.	Max	Avg.	Med.	Max	Avg.	Med.
ENm001	all	49816	127	0.36	0	255	2.12	1	74361	41.55	0	1566361	166.30	2
	>= 2 B.	13752	127	0.10	0	255	5.06	1	9205	136.15	15	1566361	568.48	22
ENm002	all	27618	78	0.38	0	354	2.10	1	59918	45.45	0	240848	121.26	2
	>= 2 B.	6799	78	0.09	0	354	5.48	1	22139	160.42	16	240848	436.78	25
ENm003	all	13919	43	0.38	0	188	2.07	1	71783	44.22	0	102444	175.65	2
	>= 2 B.	3510	43	0.10	0	188	5.24	1	11319	145.79	16	102444	640.60	24
ENm004	all	43840	120	0.39	0	1015	2.05	1	4987160	181.38	0	4987160	328.23	2
	>= 2 B.	11026	119	0.10	0	1015	5.15	1	441976	202.88	17	1756334	746.46	28
ENm005	all	47581	238	0.39	0	289	2.03	1	116021	40.62	0	1088526	127.62	2
	>= 2 B.	12139	238	0.10	0	289	5.04	1	14384	143.41	15	1088526	458.29	25



merge and undo blocks according to the set of homologies until the block set maps the input sequences sufficiently but no block has degree higher than a pre-specified maximum degree (by default a maximum degree of 50 was used at all stages).

To test this procedure we constructed multi-level cacti using the described procedure for the first five Encode Pilot Project [21] ENCODE, ENm001 (the CFTR), ENm002 (the interleukin cluster), ENm003 (the apo cluster), ENm004 (region on Chr22) and ENm005 (region on Chr21). For each region we used seven placental mammal genome sequences from Human, Chimpanzee, Baboon, Mouse, Rat, Dog and Cow; the total sizes of all the input sequences for each region ranged from between 3.5 to 22.3 million bases. On a dual-core 2.6GHz laptop with 4 gigabytes of memory alignments for each region took from between 1 to 3 hours. Table 1 gives statistics for nets in the resulting cactus.

Of particular interest is the balance of the resulting trees. Using the measure of relative entropy (see table legend) we calculate that the average number of bits required to encode a path from the root of the multi layered cactus tree to the net whose child chains contain it is between approximately 25% and 60% more than that required in an optimally balanced tree. The depth of the tree is also important in considering indexing the cactus tree, because some nets have a very high degree of branching the median and average depths of leaf nets in the multi-level cactus tree is only around 7-8.

Table 2 shows statistics on the chains in the multi-level cactus tree. We break the analysis up into two categories, firstly we consider all chains and secondly we consider only chains containing a minimum of two blocks. This is because many chains involve a single block (typically with degree 1, corresponding to an indel) and an inherited end from a parent net contained group and are therefore relatively uninteresting. We observe many long chains, in terms of block number, total combined block basepair length and average instance length of a thread running through a chain and its links. Figures S1 in the supplement shows the distribution of these length metrics, Figures S2-S4 show the relationship between these length metrics. There is a clear trend for longer chains in terms of links to have longer total basepair block lengths and longer instance lengths, however we also observe chains with few links, and therefore few blocks with very long instance lengths and block lengths. These latter chains typically correspond to either lineage specific insertions, as mentioned, or in the case where the average instance length is much larger than the total block length, to where chains span very large links. This latter case occurs where order and orientation is conserved between the very ends of the input sequences but there is substantial rearrangement in the rest of the sequences which prevent intermediate links in these chains. In the supplement Tables S1 and S2 and Figures S-7 analyze metrics of block length and end connectivity. We note that most block ends are not highly connected (average 1.5, median 1 adjacencies to other distinct ends), and that more than 90% of groups contain less than 10 block ends.

### 3 Discussion

This paper has described how cactus graphs provide a hierarchical decomposition of genomes into a series of nested chains and nets, given a homology mapping. Furthermore our implementation demonstrates that for substantial regions it is possible to construct large multi-level cactus graphs that are reasonably balanced and highly branching so that their median depth from root to leaves is short. We therefore believe that cactus graphs will prove useful for visualizing, storing, indexing and ultimately reasoning about genome comparisons. We are developing several extensions to this work along these lines. In Section 1.3 of the appendix we explore a complementary idea to that of multi-level cactus graphs, that there is often a natural hierarchical organization for subsets of input genomes, e.g. by evolutionary sub-clades. We thus propose recursively constructing (multi layer) cactus graphs progressively across sub-clades of an organizing phylogeny or lineage of events (in the case of cancer genomes), linking cacti naturally together using ancestral genomes derived from each cactus.

The supplementary material for this paper can be found at: [http://compbio.soe.ucsc.edu/reconstruction/cactus\\_recomb2010\\_paper/supplement.pdf](http://compbio.soe.ucsc.edu/reconstruction/cactus_recomb2010_paper/supplement.pdf)

### References

1. Miller, W., Rosenbloom, K., Hardison, R.C., Hou, M., Taylor, J., Raney, B., Burhans, R., King, D.C., Baertsch, R., Blankenberg, D., Pond, S.L.K., Nekrutenko, A., Giardine, B., Harris, R.S., Tyekucheva, S., Diekhans, M., Pringle, T.H., Murphy, W.J., Lesk, A., Weinstock, G.M., Lindblad-Toh, K., Gibbs, R.A., Lander, E.S., Siepel, A., Haussler, D., Kent, W.J.: 28-way vertebrate alignment and conservation track in the ucsc genome browser. *Genome Res.* 17(12), 1797–1808 (2007)
2. Paten, B., Herrero, J., Beal, K., Fitzgerald, S., Birney, E.: Enredo and pecan: Genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res.* 18(11), 1814–1828 (2008)
3. Carver, T., Thomson, N., Bleasby, A., Berriman, M., Parkhill, J.: Dnaplotter: circular and linear interactive genome visualization. *Bioinformatics* 25(1), 119–120 (2009)
4. Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S.J., Marra, M.A.: Circos: an information aesthetic for comparative genomics. *Genome Research* 19(9), 1639–1645 (2009)
5. Diskin, S.J., Hou, C., Glessner, J.T., Attiyeh, E.F., Laudenslager, M., Bosse, K., Cole, K., Mossé, Y.P., Wood, A., Lynch, J.E., Pecor, K., Diamond, M., Winter, C., Wang, K., Kim, C., Geiger, E.A., McGrady, P.W., Blakemore, A.I.F., London, W.B., Shaikh, T.H., Bradfield, J., Grant, S.F.A., Li, H., Devoto, M., Rappaport, E.R., Hakonarson, H., Maris, J.M.: Copy number variation at 1q21.1 associated with neuroblastoma. *Nature* 459(7249), 987–991 (2009)
6. Bignell, G.R., Santarius, T., Pole, J.C.M., Butler, A.P., Perry, J., Pleasance, E., Greenman, C., Menzies, A., Taylor, S., Edkins, S., Campbell, P., Quail, M., Plumb, B., Matthews, L., McLay, K., Edwards, P.A.W., Rogers, J., Wooster, R., Futreal, P.A., Stratton, M.R.: Architectures of somatic genomic rearrangement in human cancer amplicons at sequence-level resolution. *Genome Research* 17(9), 1296–1303 (2007)

7. Hampton, O.A., Hollander, P.D., Miller, C.A., Delgado, D.A., Li, J., Coarfa, C., Harris, R.A., Richards, S., Scherer, S.E., Muzny, D.M., Gibbs, R.A., Lee, A.V., Milosavljevic, A.: A sequence-level map of chromosomal breakpoints in the mcf-7 breast cancer cell line yields insights into the evolution of a cancer genome. *Genome Research* 19(2), 167–177 (2009)
8. Kent, W.J., Baertsch, R., Hinrichs, A., Miller, W., Haussler, D.: Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc. Natl. Acad. Sci. USA* 100(20), 11484–11489 (2003)
9. Harary, F., Uhlenbeck, G.: On the number of husimi trees, i. *Proceedings of the National Academy of Sciences* 39, 315–322 (1953)
10. Bergeron, A., Stoye, J.: On the similarity of sets of permutations and its applications to genome comparison. *J. Comput. Biol.* 13(7), 1340–1354 (2006)
11. Bergeron, A., Mixtacki, J., Stoye, J.: Reversal distance without hurdles and fortresses. In: Sahinalp, S.C., Muthukrishnan, S.M., Dogrusoz, U. (eds.) *CPM 2004*. LNCS, vol. 3109, pp. 388–399. Springer, Heidelberg (2004)
12. Korneyenko, N.M.: Combinatorial algorithms on a class of graphs. *Discrete Applied Mathematics*, 109–111 (1994)
13. Zamazek, B., Zerovnik, J.: Estimating the traffic on weighted cactus networks in linear time. In: *Ninth International Conference on Information Visualisation (IV 2005)*, pp. 536–541 (2005)
14. Ben-Moshe, B., Bhattacharya, B.: Efficient algorithms for the weighted 2-center problem in a cactus graph. In: Deng, X., Du, D.-Z. (eds.) *ISAAC 2005*. LNCS, vol. 3827, pp. 693–703. Springer, Heidelberg (2005)
15. Tetsuo, N.: On the number of solutions of a class of nonlinear resistive circuit. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, Singapore, pp. 766–769 (1991)
16. Alekseyev, M.A., Pevzner, P.A.: Breakpoint graphs and ancestral genome reconstructions. *Genome Research* 19(5), 943–957 (2009)
17. Pevzner, P.A., Tang, H., Waterman, M.S.: An eulerian path approach to dna fragment assembly. *Proc. Natl. Acad. Sci. USA* 98(17), 9748–9753 (2001)
18. Raphael, B., Zhi, D., Tang, H., Pevzner, P.: A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res.* 14(11), 2336–2346 (2004)
19. Tsin, Y.H.: A simple 3-edge-connected component algorithm. *Theory Comput. Syst.* 40(2), 125–142 (2007)
20. Lunter, G., Rocco, A., Mimouni, N., Heger, A., Caldeira, A., Hein, J.: Uncertainty in homology inferences: assessing and improving genomic sequence alignment. *Genome Res.* 18(2), 298–309 (2008)
21. ENCODE-Consortium: Identification and analysis of functional elements in 1 genome by the encode pilot project. *Nature* 447(7146), 799–816 (2007)

# IDBA – A Practical Iterative de Bruijn Graph De Novo Assembler

Yu Peng, Henry C.M. Leung, S.M. Yiu, and Francis Y.L. Chin

Department of Computer Science, The University of Hong Kong  
Pokfulam Road, Hong Kong  
{ypeng, cmleung2, smyiu, chin}@cs.hku.hk

**Abstract.** The de Bruijn graph assembly approach breaks reads into  $k$ -mers before assembling them into contigs. The string graph approach forms contigs by connecting two reads with  $k$  or more overlapping nucleotides. Both approaches must deal with the following problems: false-positive vertices, due to erroneous reads; gap problem, due to non-uniform coverage; branching problem, due to erroneous reads and repeat regions. A proper choice of  $k$  is crucial but for single  $k$  there is always a trade-off: a small  $k$  favors the situation of erroneous reads and non-uniform coverage, and a large  $k$  favors short repeat regions.

We propose an iterative de Bruijn graph approach iterating from small to large  $k$  exploring the advantages of the in between values. Our IDBA outperforms the existing algorithms by constructing longer contigs with similar accuracy and using less memory, both with real and simulated data. The running time of the algorithm is comparable to existing algorithms.

**Availability:** IDBA is available at <http://www.cs.hku.hk/~alse/idba/>

**Keywords:** De novo assembly, de Bruijn graph, string graph, mate-pair, high throughput short reads.

## 1 Introduction

Despite tremendous research efforts, de novo assembly remains an only partially solved problem. Although more reference genomes are known for efficient resequencing, de novo assembly remains a critical step in studying a genome. Applications such as detection of structural variations [1] cannot be done easily based on resequencing techniques and there are evidences that show genome assembly based on resequencing may produce errors especially for species with high mutation rates [1].

With high throughput sequencing technologies (e.g. Illumina Genome Analyzer and Applied Biosystems SOLiD), mate-pair short reads (35nt to 75nt) of a mammalian genome can be generated in a few weeks at low cost. As short reads have different characteristics (i.e. shorter length, higher coverage, but relatively higher error rates) when compared to traditional Sanger reads, new assembly tools have emerged [2-11]. The first batch of tools (e.g. SSAKE [3], VCAKE [4], SHARCGS [5]) uses the “overlap-then-extend” idea but need to rely on data structures such as prefix trees, so they require lots of memory and run very slowly. The newer tools are divided into those based on the de Bruijn graph (e.g. Velvet [7], Abyss [8], Euler-SR[2, 6],

AllPaths[11]) and those based on the string graph (e.g. Edena [9]). Each of the two approaches has merits and limitations, and it is not clear which is better.

*De Bruijn graph algorithms* [7-8, 12-13] assemble reads by constructing a de Bruijn graph in which each vertex represents a length- $k$  substring ( $k$ -mer) in a length- $l$  read and connects vertex  $u$  to vertex  $v$  if  $u$  and  $v$  are *consecutive  $k$ -mers* in a read, i.e. the last  $(k - 1)$  nucleotides of the  $k$ -mer represented by  $u$  is the same as the first  $(k - 1)$  nucleotides of the  $k$ -mer represented by  $v$ . Intuitively, maximal paths of vertices without branches in the graph correspond to contigs to be outputted by algorithms.

*String graph algorithms* [9, 14] represent each read by a vertex and there is a directed edge from vertex  $u$  to vertex  $v$  if the suffix of at least  $x$  nucleotides of read  $u$  is the same as the prefix of read  $v$ . The value of  $x$  is the number of overlapping nucleotides for two consecutive reads. Similar to de Bruijn graph algorithms, string graph algorithms report maximal paths without branches as a contigs.

When the reads are error-free with high coverage, most tools work well. However, because of repeats, erroneous reads, and non-uniform coverage, their performances is not always acceptable. In this paper, we focus on three major problems: (1) *false positive vertices* (due to errors in reads); (2) *gap problem* (due to non-uniform or low coverage) and; (3) *branching problem* (due to repeats or errors in reads).

### Three major problems

(a) *False Positive Vertices*: Errors in reads introduce false positive vertices which make both graphs bigger and consume more memory; for example, for the human genome with 30x coverage, the memory requirement of Velvet [7] and Abyss[8] is more than 250G.

(b) *Gap problem*: Due to non-uniform or low coverage, reads may not be sampled for every position in the genome. For the de Bruijn graph, when all the (possible  $l - k$ ) reads covering consecutive  $k$ -mers are missing, we may have short “dead-end” paths. The larger the  $k$ , the more serious is the gap problem. The same applies to the string graph if all the (possible  $l - x$ ) reads following another read are missing.

(c) *Branching problem*: Those  $k$ -mers which connect with multiple  $k$ -mers due to repeat regions or erroneous reads introduce branches in the de Bruijn graph. Many algorithms [7-9] stop the contigs at branches and it is not possible to extend a contig without additional information. A small  $k$  will lead to more branches. The same branching problem occurs in string graph algorithms, and depending on the value of  $x$ , the same read can be connected with multiple other reads.

### Existing assembly algorithms

Table 1 summarizes the major techniques used by existing algorithms to solve the above problems. There are two methods for handling false positive vertices. (1) *Dead-end removal*: False positive vertices usually lead to short dead-end paths. Both de Bruijn and string algorithms (e.g. [7-8]) remove false positive vertices by removing these paths. However, due to the gap problem, some paths may be removed by mistake. (2) *Filtering*: de Bruijn graph algorithms remove false positive vertices if the corresponding  $k$ -mers appear no more than  $m$  times. However, some correct  $k$ -mers with low coverage might also be removed especially for large  $k$  for which the expected  $k$ -mer occurrence frequency is low. As for string graph algorithms, the

expected occurrence of each read is also low (1 or 2) and they rely on error correction which falls back to the multiplicity of  $k$ -mers [15] to correct errors in each read before forming contigs.

**Table 1.** Major techniques used to handle the three problems

Problems	Techniques	de Bruijn graph	String graph
1) False positive vertices	(i) Dead-end removal	Yes	Yes
	(ii) Filtering	Yes (not effective if $k$ is large)	Not applicable (relies on error correction algorithms)
2) Gap	No effective method (try to use a reasonable small $k$ or $x$ )	--	--
3) Branching	(i) Using read information	Yes	Not applicable (already use the whole read information)
	(ii) Bubble removal	Yes	Yes

There is no effective method to deal with the gap problem except all algorithms try to avoid gaps by using a small  $k$  (or  $x$  in string graphs).

Some de Bruijn graph algorithms [12] solve the branching problem by considering only those branches that are supported by reads. However, this method may easily lead to erroneous contigs [12] if the reads are erroneous especially when error rates are high. This method cannot be applied to string graph algorithms as they already consider the read information. The other technique is *bubble removal*, which is used by both approaches [7-9] and tries to merge similar paths of very similar vertices into *one path* as the small differences may only be due to SNPs or errors. However, the merging might be incorrect and this process increases the length of contigs at the expense of their accuracy.

**Table 2.** Performance (N50) of three existing assembly algorithms (Edena, Velvet, Abyss) against IDBA under different coverage and error rates for the simulated dataset using E.coli where read length is 75nt. The best results generated are used for comparison.

	High Coverage Low Error Rate (100x, 0.5%)	Low Coverage Low Error Rate (30x, 1%)	High Coverage High Error Rate (100x, 2%)	Low Coverage High Error Rate (30x, 2%)
Edena (string Graph)	63256	5104	53491	147
Velvet (de Bruijn Graph)	63214	24772	59285	16527
Abyss (de Bruijn Graph)	58678	22109	50009	10992
IDBA (our algorithm)	63218	63218	59287	32612

To summarize, string graph algorithms do not have an effective method to remove errors from reads and have the gap problem if  $x$  is set to a reasonable value to avoid the branching problem. However, string graph algorithms, which make use of the direct information in the whole read, perform very well in case of high coverage and low error rate. For other cases, de Bruijn graph algorithms may perform better. Our observations are confirmed by the N50 comparison of Edena [9] (currently one of the best string graph algorithms), Velvet [7] and Abyss [8] (the best de Bruijn graph algorithms) based on different coverage and error rates of the data as shown in Table 2 (more details on the comparison can be found in Section 4).

The best existing assembly algorithms are Edena (string graph based), Velvet and Abyss (both de Bruijn graph based, differing in the exact details for handling dead-ends and bubble removal). However, setting the correct parameter  $k$  in de Bruijn graph algorithms (or  $x$  in string graph algorithms) is crucial. The  $k$  parameter (or  $x$ ) affect the filtering and, moreover, provides a trade-off between the gap problem and the branching problem. In order to minimize the number of gaps, a smaller  $k$  (or  $x$ ) should be used. But with a small  $k$  (or  $x$ ), the branching problem becomes more serious. Existing algorithms usually pick a moderate value for  $k$  (or  $x$ ) to balance between the two problems. None of the existing approaches try to take advantage of using different  $k$  (or  $x$ ) values<sup>1</sup>.

### Our contributions

We propose a new assembly algorithm (IDBA) based on the de Bruijn graph. The idea is simple but practical in that it alleviates the difficulties in setting a correct  $k$  and the filtering threshold  $m$ , gives good results, uses much less memory (many existing tools require huge amount of memory making them impractical for large genomes) at the expense of a reasonable increase in running time. Instead of using a fixed  $k$ , our algorithm iterates from small to large  $k$  ( $k_{\min}$  to  $k_{\max}$ ) capturing the merits of all values in between. The key step is to maintain an accumulated de Bruijn graph to carry useful information forward as  $k$  increases. Note that this is not the same as running the algorithm for many different  $k$  values independently as it is not clear how to combine contigs from different runs to get a better result. We show theoretically that the accumulated de Bruijn graph can capture good contigs and these contigs can be made longer as  $k$  increases. Based on experiments on simulated and real data, we show that IDBA can produce longer contigs (see Table 2 for the N50 comparison) with similar accuracy (very few wrong contigs and high coverage). More detailed results are presented in Section 4.

We are able to reduce the memory consumption by 50-80% as compared to existing algorithms which use a fixed  $k$  of moderate size. Because  $k$  is of moderate size, the algorithms cannot do filtering in the first step especially when the coverage is not high and thus create a big graph due to false positive vertices. However, since IDBA starts with a small  $k$ , many false positive vertices are pruned with a conservative and effective filtering in the first step (e.g., set  $m=1$ ). Although IDBA iterates through different  $k$  values, with implementation tricks (described in Section 2), IDBA runs a lot faster than Abyss and is comparable with other existing algorithms.

### Organization of the paper and remarks

We organize the paper as follows. In Section 2, we introduce our algorithm IDBA and show the advantages of using small and large  $k$  values. Also, we provide key implementation details which help to reduce the memory consumption and running time. Section 3 compares the performance of IDBA with existing algorithms on both simulated and real data. We conclude the paper in Section 4.

---

<sup>1</sup> The SHARCGS [5] algorithm uses fixed  $x$  values (the number of overlapping nucleotides) when extending a read, but they repeat the whole assembling procedure *independently* using a few different  $x$  values and combine the resulting contigs from different runs only.

We note that using mate-pair information to resolve repeats that are longer than reads is another important aspect of an assembly tool. In this paper, we mainly focus on short repeats, which account for the largest portion of repeats in genomes and cannot be resolved by mate-pair information easily as the variation of the insert size may be even larger than the length of the repeat. We leave the problem of how to use mate-pair information in assembly more effectively for future study. Hence, the last step of our assembly tool, which uses mate-pair information to connect the contigs, simply follows Abyss [8]. Note also that, although our approach can be applied to the string graph with a range of  $x$  values, currently there is no effective way to remove errors from reads for string graphs, and so we focus on de Bruijn graphs.

## 2 Algorithm IDBA

Given a set of reads, we denote the de Bruijn graph for any fixed  $k$  as  $G_k$ . Instead of using only one fixed  $k$ , IDBA (Iterative de Bruijn Graph short read Assembler) iterates on a range of  $k$  values from  $k = k_{\min}$  to  $k = k_{\max}$  and maintains an *accumulated de Bruijn graph*  $H_k$  at each iteration. In the first step,  $k = k_{\min}$ ,  $H_k$  is equivalent to the graph  $G_k$  after deleting all vertices whose corresponding  $k$ -mers appear no more than  $m$  times (we set  $m = 1$  or  $2$  in practice depending on the coverage of the input reads) in all reads. Theorem 3 (in the Appendix) shows that these  $k$ -mers are very likely to be false positives.

To construct  $H_{k+1}$  from  $H_k$ , we first construct potential contigs in  $H_k$  by identifying maximal paths  $v_1, v_2, \dots, v_p$  in which all vertices have in-degree and out-degree equal to 1 except  $v_1$  and  $v_p$  which may have in-degree 0 and out-degree 0, respectively. Note that a path of  $p$  vertices represents a potential contig of length  $p + k - 1$ . We remove all reads that can be represented by potential contigs in  $H_k$  i.e. those reads that are substrings of a contig (as these reads cannot be used to resolve any branch). In the construction of  $H_{k+1}$ , we only consider the remaining reads and the potential contigs in  $H_k$ . We perform two steps to convert  $H_k$  to  $H_{k+1}$ . (1) For each edge  $(v_i, v_j)$  in  $H_k$ , we convert the edge into a vertex (representing a  $(k+1)$ -mer  $x_{i_1} x_{i_2} \dots x_{i_k} x_{j_k} = x_{i_1} x_{j_1} \dots x_{j_k}$ ). (2) We connect every two such vertices by an edge if the corresponding two consecutive  $(k+1)$ -mers have support from one of the remaining reads or potential contigs of  $H_k$ , i.e. the corresponding  $(k+2)$ -mer exists.

Note that in practice, we do not need to go from  $k$  to  $k+1$ ; we can jump from  $k$  to  $k+s$ , in which case, for (1), we convert each path of length  $s$  in  $H_k$  into a vertex. In Theorem 5 in the Appendix, we show that by setting  $s = 1$ , we may get high quality contigs. As  $s$  increases, we expect the quality of contigs will drop, so it is always better to use a small  $s$ . The choice of  $s$  will represent a trade-off on the efficiency of the algorithm and the quality of the contigs.

For each  $H_k$ , we follow other algorithms [7] to remove dead-ends (potential contig shorter than  $3k - 1$  with one end with 0 in-degree or out-degree, which represents a path in  $H_k$  of length at most  $2k$ ). Note that removing a dead-end may create more dead-ends, the procedure will repeat until no more dead-ends exist in the graph. These dead-end contigs are likely to be false positives (to be discussed in the Appendix). In fact, most of the remaining false positive vertices after the first filtering step can be removed as dead ends and the accuracy of the contigs produced by IDBA is high.



After obtaining  $H_{k_{\max}}$ , we merge *bubbles* where bubbles are two paths representing two different contigs going from the same vertex  $v_1$  to the same vertex  $v_p$  where these two contigs differ by only one nucleotide. This scenario is likely to be caused by an error or a SNP. Like other assembly algorithms [7-9], we merge the two contigs into one. We base on mate-pair information to connect the contigs as much as possible by using a similar algorithm as Abyss[8] and report the final set of contigs.

### Algorithm IDBA

```

1   $k \leftarrow k_{\min}$  ( $k_{\min} = 25$  by default)
2  Filter out  $k$ -mers appearing  $\leq m$  times
3  Construct  $H_{k_{\min}}$ 
4  Repeat
5     a) Remove dead-ends with length  $< 2k$ 
6     b) Get all potential contigs
7     c) Remove reads represented by potential contigs
8     d) Construct  $H_{k+s}$  ( $s = 1$  by default)
9     e)  $k \leftarrow k + s$ 
10 Stop if  $k \geq k_{\max}$  ( $k_{\max} = 50$  by default)
11 Remove dead-end with length shorter than  $2k_{\max}$ 
12 Merge bubbles
13 Connect potential contigs in  $H_{k_{\max}}$  using mate-pair information
14 Output all contigs

```

Note that the probability of removing a true positive vertex in our filtering step is very low (Theorem 3 in Appendix A.3 gives the analysis) as long as  $k_{\min}$  and the filtering threshold  $m$  are set to a reasonable value (e.g.  $m = 1$ ). For example, if  $1.6 \times 10^6$  length-75 reads are sampled from a genome of length  $4.1 \times 10^6$  (45x coverage) with error rate 1%, the probability of filtering out a true positive vertex in  $H_{25}$  is  $1.14 \times 10^{-9}$ , i.e. the expected number of false negative vertices is  $0.0047 \ll 1$  which is very small. Even for some cases where the expected number of false negative vertices is large, say 10, it is still relatively very small when compared with the genome size. Thus, for simplicity in analysis, we assume there is no false negative vertex in  $H_{k_{\min}}$ . The filtering step can remove a large portion of the false positive vertices. Most of the remaining false positive vertices are removed in later steps by dead-ends. The probability of removing a correct contig as a dead-end is also small (see Theorem 4 in Appendix A.3 for the exact calculation of the probabilities). The probability of determining a dead-end wrongly is only  $2.46 \times 10^{-4}$  when the above example is considered.

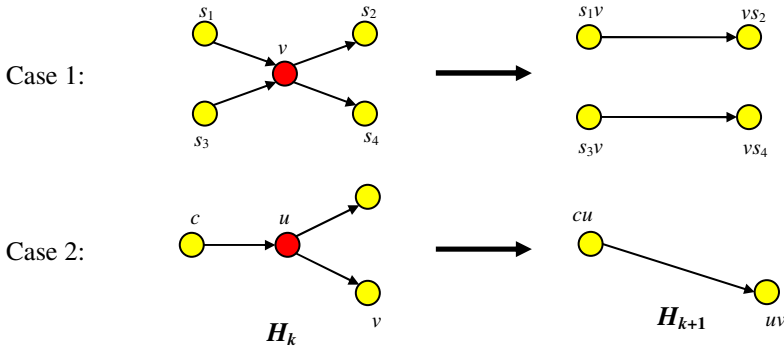
Due to the gap problem a contig that appears in  $G_k$  for a small  $k$ , might not be a contig in  $G_{k'}$  for  $k' > k$ . However, in IDBA, if a contig  $c$  appears in  $H_k$ , there must be a contig  $c'$  in  $H_{k'}$  containing  $c$  (Theorem 1). That is, the contig information is carried over from  $H_k$  to  $H_{k'}$ . As  $k$  increases, more branches can be resolved while the gaps solved when  $k$  is small in previous iterations will be preserved.

**Theorem 1:** Assume that  $k_{\min} = k$  and  $k < k'$ . If there is a contig  $c$  in  $G_k$  of length at least  $3k_{\max} - 1$  with all true positive vertices, there must be a contig  $c'$  in  $H_{k'}$  such that  $c'$  contains  $c$ .

*Proof:* By induction on  $k$ . Let  $k' = k + 1$  and  $c = x_1x_2\dots x_{p+k-1}$  be a contig in  $H_k$  represented by the path  $p = (v_1, v_2, \dots, v_p)$ , all vertices  $v_1, v_2, \dots, v_p$  have in-degree and out-degree  $\leq 1$ , it is easy to see that the path  $p' = (v_1', v_2', \dots, v_{p-1}')$  in  $H_{k+1}$  where each  $(k+1)$ -mer  $v_i' = x_ix_{i+1}\dots x_{i+k}$  also has in-degree and out-degree  $\leq 1$ . As the length of the contig represented by path  $p' \geq 3k_{\max} - 1$ , there must be a contig including path  $p$ , i.e.  $c$ , in  $H_{k+1}$ .  $\square$

**Corollary:**  $H_{k_{\max}}$  must contain all contigs in  $G_{k_{\min}}$  of length at least  $3k_{\max} - 1$  with all true positive vertices.

In practice  $H_{k_{\max}}$  always contains longer contigs than  $G_{k_{\min}}$  by resolving branches at each iteration. As Figure 1 shows, by iterating the graph  $H_k$  towards larger  $k$ , we may get longer and longer contigs as some of the branches (e.g. length- $k$  repeat region (Case 1) and error branches in  $H_{k+1}$  (Case 2)) may be resolved when using a larger  $k$ .



**Fig. 1.** Two cases for having longer contigs

Case 1: Let  $c_1 = s_1v_r s_2$  and  $c_2 = s_3v_r s_4$  be two substrings in the genome where  $v_r$  is a common length- $k$  substring representing a repeat region,  $s_1, s_2, s_3, s_4$  are different substrings.  $c_1$  and  $c_2$  are represented by five contigs in  $H_k$  as the  $k$ -mer  $v_r$  has in-degree of 2 and out-degree of 2. If there are two correct reads containing  $v_r$  and its 2 neighboring nucleotides at both ends in  $c_1$  and  $c_2$  respectively, and there is no error read containing  $s_1v_r s_4$  or  $s_3v_r s_2$ , then there must be two contigs, one containing  $c_1$  and the other containing  $c_2$  in  $H_{k+1}$ .

Case 2: Let  $c$  be a contig in  $H_k$  that stops before vertex  $u$  whose in-degree is 1 and out-degree is  $>1$ . Assume that among all branches of  $u$ , only  $u$  to  $v$  is correct. If there is a correct read containing  $u$  and its 2 pairs of neighboring nucleotides at both ends and there is no error read linking  $c$  with other branches, there will be a longer contig  $c'$  in  $H_{k+1}$  that contains  $c$ .

Case 1 and Case 2 prove the following theorem.

**Theorem 2:** If there is a contig  $c$  in  $G_k$  of length at least  $3k_{\max} - 1$  with all vertices are true positive which satisfies case 1 or case 2 in  $H_k$ ,  $k = k_{\min} \leq k' < k_{\max}$ , there is a longer contig  $c'$  in  $H_{k_{\max}}$  that contains  $c$ .

In the algorithm, we increase the value of  $k$  by 1 at each iteration, i.e.  $s = 1$ . Theorem 5 in Appendix A.3 shows that for a better quality of the contigs, this is essential. On the other hand, as a trade-off between the efficiency of the algorithm and the quality of the contigs, it is possible to set  $s > 1$ , i.e. to increase the value of  $k$  by more than 1 at each iterative step.

## 2.1 Implementation Details

The memory used by IDBA is only about 20-30% of that used by the other existing tools because 80% of false positive vertices are removed in the filtering step (line 2 in algorithm IDBA) and IDBA uses a compact hash table to represent de Bruijn graph implicitly with each edge represented by one bit only.

Although IDBA constructs  $H_{k_{\max}}$  from  $H_{k_{\min}}$  step by step, the running time of IDBA is not directly proportional to the number of  $k$  values between  $k_{\max}$  and  $k_{\min}$ . According to Theorem 1, a contig in  $H_k$  is also a contig in  $H_{k+1}$ , thus IDBA only needs to check whether a branch in  $H_k$  can be resolved in  $H_{k+1}$ . Since reads represented by a contig are removed in each iteration, the number of reads in each iteration decreased. In practice, about half of the reads are removed when constructing  $H_{k_{\min+1}}$  and IDBA runs much faster than Abyss, and about three times slower than Velvet.

## 3 Experimental Results

The genome of *Escherichia coli* (O157:H7 str. EC4115) from NCBI [16] is used for simulated experiments (the genome length is 5.6 M). Reads are randomly sampled uniformly with coverage 30x. In our experiments, we generated reads with error rates 1%, read length 75 and insert distance 250. Note that we have repeated the experiments using other coverage (e.g. 50x, 100x), error rates (e.g. 2%) and read length (e.g. 50). The results are similar, so we only show the result for 30x coverage with 1% error on length-75 reads. We also use a real data set, namely *Bacillus Subtilis*, to evaluate our algorithm. The length of the genome is 4.1M. The reads are sequenced using Solexa machine with coverage 45x, read length 75 and insert distance 400. The estimated error rate is about 1%.

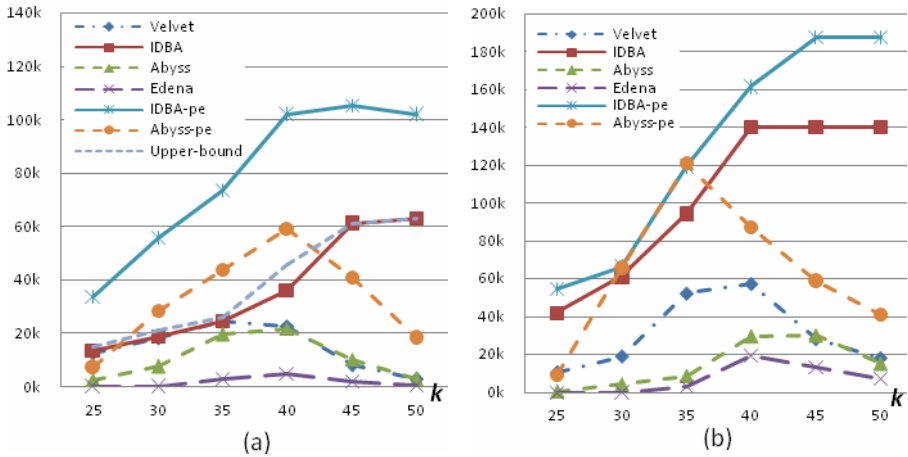
### 3.1 Simulated Data

We compare the performance of Velvet<sup>2</sup>, Abyss<sup>3</sup>, Edena and our algorithm IDBA on the simulated data based on different  $k$  values (or  $x$  values). For IDBA, we fix  $k_{\min} = 25$ ,  $m = 1$  and compare the performance of IDBA with different  $k_{\max}$ . For the other algorithms, default parameters provided by the assemblers are used. We also plot the upper bound that can be achieved by building an ideal de Bruijn graph with no false positive or false negative vertices and edges and produce all single paths as contigs.

<sup>2</sup> Since pair-end version of Velvet performs not well and pair-end version of Abyss outperforms Velvet in the quality of results, we only show result of single-end version of Velvet.

<sup>3</sup> Since SHARCGS is too slow and Abyss applies a similar idea as Euler with better performance, we leave SHARCGS and Euler out of our comparison.

We calculate N50 and coverage only for valid contigs which are longer than 100 bps and can be aligned to the reference with 99.9% similarity. Figure 2(a) shows the comparison of the algorithms based on N50. As we mentioned in the introduction (Section 1), existing assembly algorithms have many false positive vertices and branching problems when  $k$  is small and they have many gaps when  $k$  is large. Thus these algorithms have the best performance (largest N50) for in-between values of  $k$  (the optimal  $k$  for Velvet, Edena and Abyss are 35, 40 and 40 respectively in this data set). Since IDBA considers a range of  $k$  values, its performance is better than the others even when considering a range of 10 values for  $k$  ( $k_{\min} = 25$  and  $k_{\max} = 35$ ). Furthermore, when IDBA considers a larger range for  $k$  ( $k_{\min} = 25$  and  $k_{\max} = 50$ ), its performance is close to the upper bound. We have only 10 false positive contigs when setting  $k_{\min} = 25$  and  $k_{\max} = 50$  while Abyss, Velvet and Edena produce 489, 19 and 650 false positive contigs respectively.



**Fig. 2.** (a) N50 for contigs produced by assembly algorithms with different  $k$ -values ( $x$ -values if the software is string graph based) on simulated data using E.coli as the reference genome where read length is 75nt, coverage is 30x and error rate is 1%. (IDBA-pe and abyss-pe are the results for using mate-pair information to extend the contigs while Edena does not use mate-pair information) (b) N50 for contigs produced by assembly algorithms with different  $k$  (or  $x$ ) values on real data from bacillus subtilis where read length is 75nt, coverage is 45x and error rate is 1%.

For IDBA and Abyss, we also apply the mate-pair information to connect the resulting contigs to make them longer. The results are shown in the same graph (IDBA-pe and abyss-pe). Note that as  $k$  increases, the N50 may drop when applying the mate-pair procedure since more branches have been resolved incorrectly and some short contigs are removed as dead-ends. In fact, further research is required on how to use mate-pair information effectively for assembly. The pair-end version of Abyss has optimal result when  $k$  is 35 while IDBA has optimal result when  $k$  is 45.

**Table 3.** Statistics of optimal (w.r.t. N50) result of each algorithm for simulated data

	Time	Memory	$k$	Number	N50	Contigs		Coverage
						Max length	False pos. contigs (total len.)	
Velvet	155s	1641M	35	1412	24772	127265	70(35589)	95.29%
Edena	957s	678M	40	4672	5104	46908	650 (72019)	97.22%
IDBA	371s	360M	25–50	1563	63218	217365	9(4654)	97.96%
IDBA-pe	412s	360M	25–45	709	105579	217365	43 (164120)	93.94%
Abyss	1114s	1749M	40	1390	22109	87118	66 (34998)	95.05%
abyss-pe	1237s	1749M	40	484	59439	226626	186 (352437)	91.39%
upper-bound	--	--	50	1561	63218	217365	0 (0)	99.11%

Table 3 shows a comprehensive statistics on the performance of the algorithms on their optimal  $k$  values (w.r.t. N50). IDBA produced much longer contigs than all other algorithms. When mate-pair information is not available, the N50 of IDBA (63218) is about three times that of the next best algorithm (24772 by Velvet) and is the same as the upper bound. IDBA also produced the fewest number of wrong contigs (a contig which cannot be aligned to the reference genome with 99.9% similarity) and the total length of all wrong contigs is only about 4500nt which is much less than the other algorithms. The coverage of IDBA is also the best among all algorithms. Since IDBA performs well on assembling single end reads, it outperforms other algorithms even when use mate-pair information. To conclude, IDBA outperforms other algorithms substantially and produces much longer contigs with higher accuracy.

### 3.2 Real Data

Figure 2(b) shows the N50 of the contigs produced by Velvet, Abyss, Edena and our algorithm IDBA on the real reads from *Bacillus Subtilis* using different  $k$  values ( $x$  values). Since the reads may not be uniformly sampled in the real data set, we use a smaller  $k_{min}$  (20nt) and keep  $m = 1$  to run IDBA. For the other algorithms, we use their default parameters except for  $k$ . We do not have the reference genome to check if a contig is valid. We calculate the N50 for all reported contigs longer than 100bp. Note that the result may not be accurate, because some algorithms may produce longer but invalid contigs. The results are consistent with that of the simulated data. Velvet, Edena and Abyss get their best performance when  $k = 40$ , 40 and 45 respectively. IDBA can keep improving the result while  $k_{max}$  is increasing.

In this data set, mate-pair information is not so useful for IDBA because using read information can already solve most of the branches. When using  $k_{max}$  equal to 50, the N50 pair-end version of IDBA produced is 30% longer than single end version. The performance of mate-pair version Abyss has similar performance as in simulated data. Its optimal  $k$  is 35, and the longest N50 it produces is even shorter than single end IDBA. In conclusion, IDBA produced the longest contigs among all algorithms. A detailed comparison is given in Table 4 in Appendix A.2.

### 3.3 Running Time and Memory Consumption

Other than Abyss (12.8 minutes – 7 hours for simulated data and 10 minutes – 1.2 hours for real data depending on the value of  $k$ ), the running time of other algorithms are more or less the same. Abyss runs much slower when  $k$  is small, probably due to its slow procedure for dealing with graphs with many false positive vertices. Velvet

(120 – 220 seconds for simulated data and 130 – 200 seconds for real data) is the fastest among all algorithms. IDBA (180 – 350 seconds for simulated data and 280 – 330 seconds for real data) runs faster than Abyss and is about three times slower than Velvet. Refer to Figures 3 and 4 in the Appendix for details.

The memory consumption is about the same for different  $k$  values across the existing algorithms. Abyss and Velvet require about 2G bytes of memory for simulated data and 1G memory for real data. IDBA only requires about 400M and 300M respectively because 80% of false positive vertices are removed in the first filtering step. Note that only 8 25-mers are removed incorrectly in simulated data set (it matches with expected number 8.88 calculated in Theorem 3). So, the memory consumption of IDBA is only about 20 – 30% of the existing de Bruijn graph tools. Edena consumes less memory than Abyss and Velvet because the number of reads is small, but still double the size used by IDBA. Refer to Tables 3 in Section 3.2 and Table 4 in the Appendix for details.

## 4 Conclusions

Our IDBA algorithm, based on de Bruijn graphs, can capture the merits of all  $k$  values in between  $k_{\min}$  and  $k_{\max}$  to achieve a good performance in producing long and correct contigs. Because the initial filtering step removes many false positive  $k$ -mers and the number of reads considered at each iterative step is reduced, the required memory and running time is much reduced. Though an accumulated de Bruijn graph is maintained at each iterative step, the running time is comparable with the existing algorithms. In fact, this running time can be further reduced if, say, one or two  $k$  values are skipped at each iterative step. In practice, the quality of the result is only slightly affected by the skipping of values, in exchange for shorter running time.

Our next target is to investigate how to better use mate-pair information for resolving long repeats in order to produce even longer and more accurate contigs.

## References

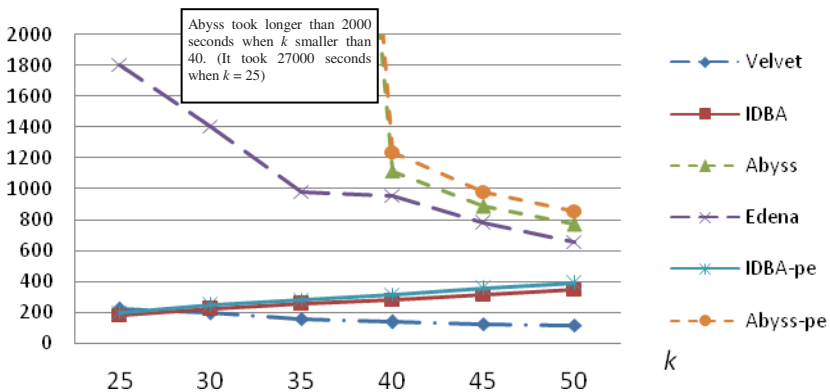
1. Wang, J., et al.: The diploid genome sequence of an Asian individual. *Nature* 456(7218), 60–65 (2008)
2. Chaisson, M.J., Brinza, D., Pevzner, P.A.: De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome Res.* 19(2), 336–346 (2009)
3. Warren, R.L., et al.: Assembling millions of short DNA sequences using SSAKE. *Bioinformatics* 23(4), 500–501 (2007)
4. Jeck, W.R., et al.: Extending assembly of short DNA sequences to handle error. *Bioinformatics* 23(21), 2942–2944 (2007)
5. Dohm, J.C., et al.: SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res.* 17(11), 1697–1706 (2007)
6. Chaisson, M.J., Pevzner, P.A.: Short read fragment assembly of bacterial genomes. *Genome Res.* 18(2), 324–330 (2008)
7. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18(5), 821–829 (2008)
8. Simpson, J.T., et al.: ABySS: a parallel assembler for short read sequence data. *Genome Res.* 19(6), 1117–1123 (2009)

9. Hernandez, D., et al.: De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. *Genome Res.* 18(5), 802–809 (2008)
10. Chaisson, M., Pevzner, P., Tang, H.: Fragment assembly with short reads. *Bioinformatics* 20(13), 2067–2074 (2004)
11. Butler, J., et al.: ALLPATHS: de novo assembly of whole-genome shotgun microreads. *Genome Res.* 18(5), 810–820 (2008)
12. Pevzner, P.A., Tang, H., Waterman, M.S.: An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. USA* 98(17), 9748–9753 (2001)
13. Idury, R.M., Waterman, M.S.: A new algorithm for DNA sequence assembly. *J. Comput. Biol.* 2(2), 291–306 (1995)
14. Myers, E.W.: The fragment assembly string graph. *Bioinformatics* 21(suppl. 2), ii79–ii85 (2005)
15. Chin, F.Y., et al.: Finding optimal threshold for correction error reads in DNA assembling. *BMC Bioinformatics* 10(suppl. 1), S15 (2009)
16. <http://www.ncbi.nlm.nih.gov/>

## Appendix

### A.1 Running Times of the Assembly Algorithms

Figure 3 and Figure 4 show the running time of IDBA and existing assembly algorithms, Velvet, Abyss and Edena on the simulated data set and the real data set. From the figures, we can see that IDBA has similar running time as other assembly algorithms except Abyss which takes a very long time when  $k$  is small due to a complicated method for removing dead-ends.



**Fig. 3.** Running time of assembly algorithms with different  $k$  (or  $x$ ) values on simulated data

### A.2 Detailed Comparison of the Assembly Algorithms for Real Data

In Table 4, we show comprehensive statistics on the performance of the algorithms on their optimal  $k$  value (w.r.t. N50) for the real dataset. IDBA produced much longer contigs than all other algorithms no matter whether the single-end or the pair-end version is used. The result is consistent with that of the simulated dataset.

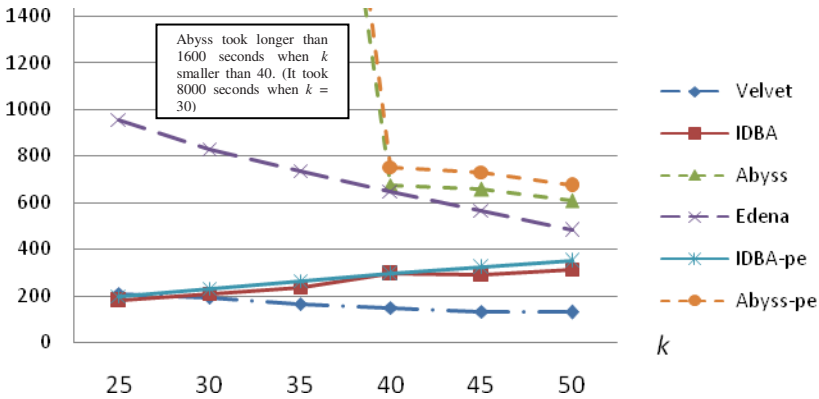


Fig. 4. Running time of assembly algorithms with different  $k$  (or  $x$ ) values on real data

Table 4. Statistics for the optimal (w.r.t. N50) result of each algorithm for real data

	Time	Memory	$k$	Total No.	Contigs	
					N50	Max length
Velvet	150s	893M	40	335	57656	181399
Edena	649s	632M	40	926	19423	66455
IDBA	325s	310M	25 – 50	267	140067	602412
IDBA-pe	361s	310M	25 – 50	203	187648	613166
Abyss	729s	923M	40	445	30081	134067
Abyss-pe	3766s	936M	35	406	120807	537397

### A.3 Theorems and Proofs

**Theorem 3:** Assume  $m$  is the filtering threshold, the probability that a  $k_{\min}$ -mer  $v$  in the genome (except the first  $l - k_{\min}$  and last  $l - k_{\min}$   $k_{\min}$ -mer in the whole genome) does not appear in  $H_{k_{\min}}$  (false negative) when  $t$  length- $l$  reads are uniformly sampled from a length- $g$  genome with error rate  $e$  is at most  $\sum_{i=0}^m \binom{t}{i} p^i (1-p)^{t-i}$  where  $p = [(l - k_{\min} + 1) / (g - l + 1)] \cdot (1 - e)^{k_{\min}}$ .

*Proof:*

$\Pr(v$  is sampled in a read)

$= \Pr(\text{read contains } v \text{ is sampled}) \Pr(v \text{ is sampled} \mid \text{read contains } v \text{ is sampled})$

$+ \Pr(\text{read does not contain } v \text{ is sampled}) \Pr(v \text{ is sampled} \mid \text{read does not contain } v \text{ is sampled})$

$\geq \Pr(\text{read contains } v \text{ is sampled}) \Pr(v \text{ is sampled} \mid \text{read contains } v \text{ is sampled})$

$\geq \frac{l - k_{\min} + 1}{g - l + 1} \cdot (1 - e)^{k_{\min}}$

The probability that a correct  $k_{\min}$ -mer  $v$  appears no more than  $m$  times is at most

$$\sum_{i=0}^m \binom{t}{i} p^i (1-p)^{t-i} \text{ where } p = \frac{l - k_{\min} + 1}{g - l + 1} \cdot (1 - e)^{k_{\min}}$$

□



**Theorem 4:** Assume that a contig  $c$  in  $H_k$  is treated as dead-end and removed. The probability that  $c$  is a correct contig is less than

$$2 \left[ 1 - \frac{l-k-2}{g-l+1} \cdot (1-e)^{k+3} \right]^l$$

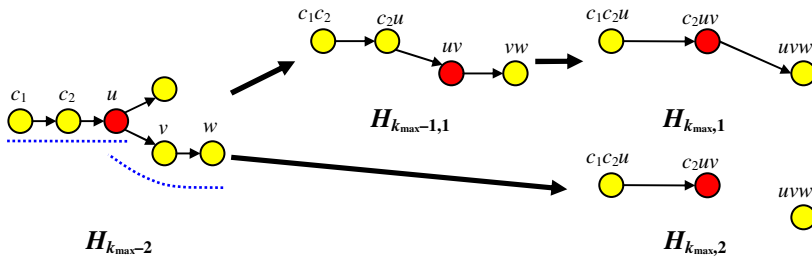
*Proof:* A contig  $c$  in  $H_k$  is treated as dead-end only if  $c$  is of length less than  $3k - 1$  and is not a dead end in  $H_{k-1}$ . Since all contigs in  $H_{k-1}$  are preserved in  $H_k$ ,  $c$  is removed because (1) the length of  $c$  is at least  $3(k - 1) - 1$  and shorter than  $3k - 1$ , or (2)  $c$  is shorter than  $3(k - 1) - 1$  and one of its ends has 0 in-degree or out-degree in  $H_k$ . Thus  $c$  will not be treated as a dead-end if the two adjacent  $(k+3)$ -mers of  $c$  is sampled. By considering  $k_{\min} = k + 3$  and  $m = 0$  in Theorem 1, the probability that no read contains a particular  $(k+3)$ -mer is at most

$$\left[ 1 - \frac{l-k-2}{g-l+1} \cdot (1-e)^{k+3} \right]$$

and the probability that no read contains a particular  $(k+3)$ -mer cover a particular end of  $c$  is at most

$$2 \left[ 1 - \frac{l-k-2}{g-l+1} \cdot (1-e)^{k+3} \right]^l$$

□



**Fig. 5.** Cases that a longer contig in  $H_{k_{\max},1}$  does not exist in  $H_{k_{\max},2}$

Let  $H_{k,s}$  denote the accumulated de Bruijn graph  $H_k$  with step size  $s$ . Theorem 5 shows that  $H_{k_{\max},1}$  has at most the same number of gaps as  $H_{k_{\max},2}$ . There are some cases (Figure 5) that there is a longer contig in  $H_{k_{\max},1}$  which is not in  $H_{k_{\max},2}$ . For example, consider a contig  $c$  in  $H_k$  which stops before vertex  $u$  whose in-degree = 1 and out-degree  $>1$  and all branches of  $u$  are shorter than  $2k$  and only  $u$  to  $v$  is correct. If there is only two reads contains  $u$  and its 2 pairs of neighboring nucleotides at two ends respectively and there is no error read linking  $c$  with other branches, there is a longer contig  $c'$  in  $H_{k+2,1}$  that contains  $c$  which does not appear in  $H_{k+2,2}$ .

**Theorem 5:** If a  $k_{\max}$ -mer ( $k_{\max+1}$ -mer) in the genome appears in  $H_{k_{\max},2}$ , it also appears in  $H_{k_{\max},1}$ .

*Proof:* By induction on  $k_{\max}$ . Consider  $k_{\max} = k_{\min} + 2$ . Given a  $k_{\max}$ -mer ( $(k_{\max}+1)$ -mer)  $v$  does not appear in  $H_{k_{\max},1}$ , let  $v'$  be the shortest substring of  $v$  of length- $k$  which does not appear as a vertex in  $H_{k,1}$  or an edge in  $H_{k-1,1}$ ,  $k_{\min} \leq k \leq k_{\max}$ .

Case 1:  $k = k_{\min}$ , i.e.  $v'$  does not appear in  $H_{k_{\min},2}$ ,  $v$  does not appear in  $H_{k_{\max}+1,2}$ .

Case 2:  $k = k_{\min}+1$ , there are two cases: (a)  $v'$  does not appear in  $H_{k_{\min},1}$  as an edge or (b)  $v'$  is a vertex on a dead-end with length less than  $2(k_{\min}+1)$  in  $H_{k_{\min}+1,1}$ . In case (a), since any  $(k_{\min}+2)$ -mer contains  $v'$  as substring does not appear in  $H_{k_{\max},2}$ ,  $v$  does not appear in  $H_{k_{\max},2}$ . In case (b),  $v$  is a vertex on a dead-end with length less than  $2(k_{\min}+2)$  in  $H_{k_{\max},2}$  which will be removed.

Case 3:  $k = k_{\min}+2$ , there are two cases: (a)  $v$  does not appear in  $H_{k_{\min}+1,1}$  as an edge or (b)  $v$  is a vertex on a dead-end with length less than  $2(k_{\min}+2)$  in  $H_{k_{\min}+2,1}$ . In case (a), consider the path  $(v_1, v_2, v_3)$  in  $H_{k_{\min},1}$  representing the  $(k_{\min}+2)$ -mer  $v$ . Since  $v$  does not appear in  $H_{k_{\min}+1,1}$  as an edge,  $v_2$  has  $>1$  in-degree or out-degree and there is no read containing  $v'$  as substring. Thus the in-degree and out-degree of  $v$  are 0 in  $H_{k_{\max},2}$  and  $v$  will be removed as dead-end. In case (b),  $v$  is a vertex on a dead-end with length less than  $2(k_{\min}+2)$  in  $H_{k_{\max},2}$  which will be removed.

Case 4:  $k = k_{\min}+3$ , i.e.  $v$  does not appear in  $H_{k_{\max},1}$  as an edge, the path  $(v_1, v_2, v_3, v_3)$  in  $H_{k_{\min},1}$  representing the  $(k_{\min}+3)$ -mer  $v$  is not a potential contig and there is no read containing  $v$  as a substring. Thus  $v$  does not appear in  $H_{k_{\max},2}$  as an edge.  $\square$

# Predicting Nucleosome Positioning Using Multiple Evidence Tracks

Sheila M. Reynolds<sup>1</sup>, Zhiping Weng<sup>2</sup>, Jeff A. Bilmes<sup>1</sup>,  
and William Stafford Noble<sup>1</sup>

<sup>1</sup> University of Washington, Seattle, Washington, USA

<sup>2</sup> Boston University, Boston, Massachusetts, USA

**Abstract.** We describe a probabilistic model, implemented as a dynamic Bayesian network, that can be used to predict nucleosome positioning along a chromosome based on one or more genomic input tracks containing position-specific information (evidence). Previous models have either made predictions based on primary DNA sequence alone, or have been used to infer nucleosome positions from experimental data. Our framework permits the combination of these two distinct types of information. We show how this flexible framework can be used to make predictions based on either sequence-model scores or experimental data alone, or by using the two in combination to interpret the experimental data and fill in gaps. The model output represents the posterior probability, at each position along the chromosome, that a nucleosome core overlaps that position, given the evidence. This posterior probability is computed by integrating the information contained in the input evidence tracks along the entire input sequence, and fitting the evidence to a simple grammar of alternating nucleosome cores and linkers. In addition to providing a novel mechanism for the prediction of nucleosome positioning from arbitrary heterogeneous data sources, this framework is also applicable to other genomic segmentation tasks in which local scores are available from models or from data that can be interpreted as defining a probability assignment over labels at that position. The ability to combine sequence-based predictions and data from experimental assays is a significant and novel contribution to the ongoing research regarding the primary structure of chromatin and its effects upon gene regulation.

**Keywords:** Nucleosome prediction, dynamic Bayesian network, chromatin structure.

## 1 Introduction

DNA in eukaryotes is packaged with histone and other proteins into a chromatin complex. The most basic element of chromatin is the nucleosome “core”, which consists of a bundle of eight histone proteins around which is wound approximately 147 base pairs (bp) of double-stranded DNA. Between adjacent cores exists a variable-length stretch of DNA commonly called the “linker” which is

generally more accessible to elements such as transcription factors than the compacted DNA in the core. The precise positioning of the nucleosome cores and the inter-nucleosomal linker regions allows for selective access to the DNA by the cellular machinery; understanding the mechanisms that control this positioning is therefore crucial to our understanding of gene regulation and expression.

Numerous computational approaches to inferring nucleosome positions either from experimental data or from the primary DNA sequence have been published in recent years. These methods generally use a hidden Markov model (HMM) or similar framework (*e.g.* Boltzmann chain) in which a sequence of hidden states, representing the nucleosome core and the linker, form a Markov chain, and the observations “emitted” by each state are derived either from DNA-sequence models or experimental assays. Common model assumptions include the requirement that adjacent nucleosomes may not overlap, as well as constraints on the length of a nucleosome and a model of the linker lengths. The model of linker lengths generally specifies a minimum linker length due to steric hindrance between adjacent nucleosomes, and may also define a geometric or other distribution over longer linker lengths [1] or an upper limit on linker length [2]. Although very similar in implementation, models based on DNA-sequence scores and models based on experimental data are solving two different problems. When the inputs to the HMM are sequence-model scores [1,2,3,4,5], the HMM framework predicts the most probable nucleosome positions based on the DNA sequence alone. In contrast, when the inputs originate from experimental data such as tiling microarrays [6,7,8,13], the goal is data analysis and interpretation.

In this work, we exploit the power of dynamic Bayesian networks (DBNs) to create a general framework for predicting nucleosome positions using one or more input tracks of arbitrary position-specific genomic scores. A DBN is a generalization of the widely used HMM [9], and generalized versions of the standard inference algorithms commonly applied to HMMs exist for the broader class of DBNs. The typical HMM falls into the broad class of generative models in that, in addition to being used in the standard way, the model can also be (although rarely is) used to generate instances of evidence sequences according to the model parameters. The model that we present here is more discriminative in nature, and uses the input evidence to directly inform the probabilities at each state in the Markov chain. Furthermore, our model allows multiple evidence tracks to be combined to jointly influence the current state, while the Markov chain simultaneously enforces the sequential grammar that is described by the state transition matrix. Specifically, we show how we can use either sequence-model scores or experimental data independently, or both together, with the result that the sequence scores can be used to fill in gaps in the experimental data and provide a more complete picture of the nucleosome landscape. Alternatively, sequence-model scores can be used in conjunction with transcription factor (TF) binding probabilities, resulting in a competitive model similar to the one described by Wasson and Hartemink [5] with the assumption that a TF can only bind to the DNA between nucleosome cores. The ability to combine

sequence-based predictions and data from experimental assays is a significant and novel contribution to the ongoing research regarding the primary structure of chromatin and its effects upon gene regulation.

## 2 Results

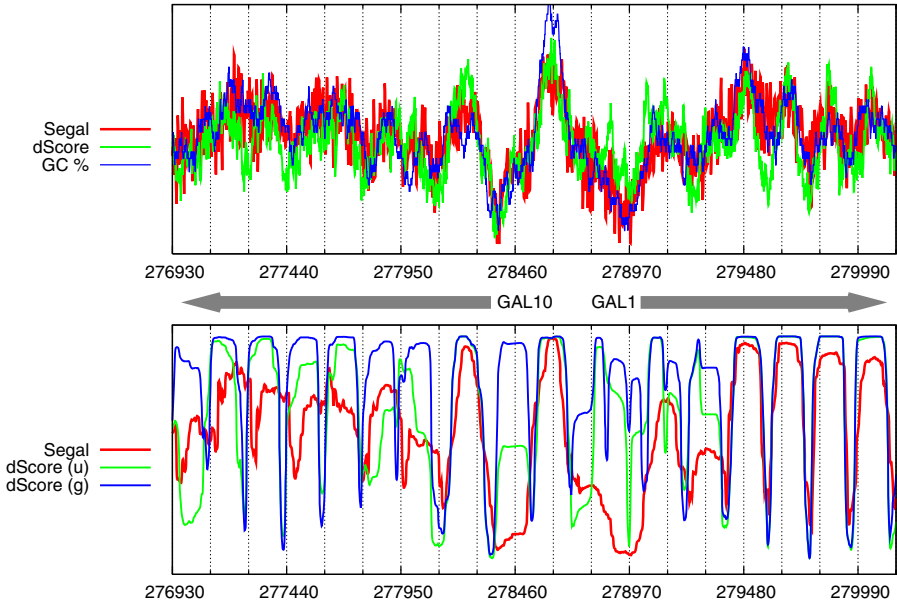
### 2.1 Predicting Nucleosome Positioning from Arbitrary Sequence-Preference Scores

In recent years, numerous methods have been proposed for scoring a DNA segment for the purposes of distinguishing nucleosome-inhibiting vs nucleosome-forming regions. The DBN presented in this work can integrate the information contained in these types of local sequence scores, regardless of the method used to produce them, to infer nucleosome positioning along a chromosome. In this section we illustrate this application of our method with three specific examples. First we show that we can recapitulate the average occupancy predicted by the Segal model [3,12,14] using the Segal raw binding scores as inputs, and then we show predictions based on our recently developed nucleosome dyad scores using two different linker-length models. Our probabilistic framework permits two types of linker models: a geometric length-distribution which prefers shorter linkers, or a uniform distribution which gives the same probability to all possible linker lengths (see *Methods* for details). These two different linker-length models can be thought of as describing two variations on the statistical positioning idea [10] in regions where sequence-directed positioning is weak.

Our nucleosome dyad score, *dScore*, is based on a discriminative pattern-correlation method [11] which computes a score for the central position of an input sequence of length 301 bp, based on sequence information alone, by weighting and combining information from all  $k$ -mers for  $k \in \{1, 2, 3\}$ . This score is the continuous-valued output of a binary classifier and can be interpreted in a manner similar to a log-ratio. The Segal raw binding score is the log-ratio of two model components: one captures the periodic positioning of dinucleotides along the nucleosome core, while the other encodes the relative linker-region preferences for all 5-mers.

Figure 1 shows the two different sequence-preference scores in the top panel: the Segal raw binding score and our dyad score (dScore), plus a GC-content track for reference (computed using a sliding window of width 71 bp). In the bottom panel, each trace corresponds to the posterior probability that a position is covered by a nucleosome core, inferred by the model from the input local sequence scores. The output based on the Segal raw binding score and using the uniform linker-length model closely recapitulates the average occupancy probability predicted by the full Segal model [12] (Pearson correlation  $r = 0.96$ ). Two separate output traces are shown based on the dScores: the first uses the uniform linker-length model, and the second uses the geometric linker-length model.

There are significant qualitative similarities as well as differences both between the Segal and dScore sequence-scores and the posterior probabilities shown in Figure 1. These differences are due to the differences in the input scores as well as



**Fig. 1.** *S. cerevisiae* chromosome II: raw sequence-model scores and local GC % (bottom) and nucleosome core posterior probabilities (top) for the Segal model and our pattern-classification model with a uniform linker model (green) and a geometric linker model (blue)

to differences in the linker length models. The most striking difference can be seen immediately upstream of the GAL10 transcription start site, in an AT-rich region wide enough for one nucleosome core, where both sequence-models produce low scores. The Segal model predicts a very long nucleosome-free region (NFR), while the two dScore models predict a weakly-positioned nucleosome—the model that prefers shorter linker lengths places a nucleosome with high probability while the uniform linker-length model places one with lower probability.

## 2.2 Interpretation of Experimental Data Alone or in Conjunction with Sequence Scores

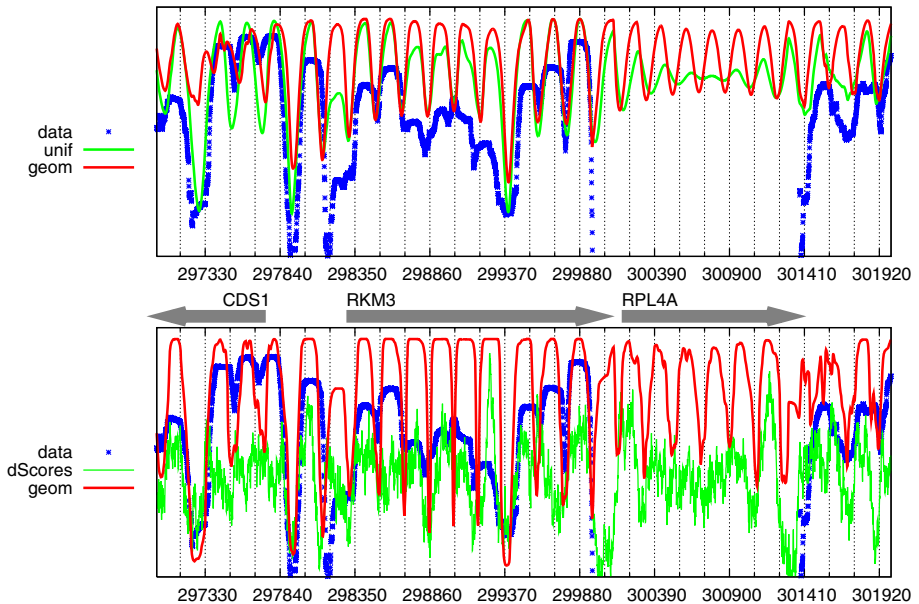
Another application of our model is to interpret experimental data, similar to what has been done previously with microarray data [6,8,13]. By incorporating additional information in the form of sequence-based scores or even just a model of linker lengths, the model can fill in gaps in the experimental data. Experimental data is frequently also expressed as a log-ratio, so the same mapping to probabilities described above can be used here.

Figure 2 shows a region on yeast chromosome II for which there is a gap in one of the *in vivo* experimental data sets from Kaplan *et al.* [12]. The gap is 1340 bp wide and corresponds to the ribosomal protein RPL4A. Using the

experimental data as an evidence track, the probabilistic model was run twice—once using the geometric linker model, and once using the uniform linker model (top panel of Figure 2). When the model includes a preference for shorter linker lengths, it places 8 nucleosomes, evenly distributed across the 1340 bp gap in the data. With the uniform linker model, we observe two interesting changes in the predictions: first, they track the input data much more closely because, aside from the grammar, the data is the only source of information; and second, the model is much less certain about how many nucleosomes fill the gap—without the preference for short linkers, the model is considering all possible placements of between one and eight nucleosomes. In both cases the uncertainty grows with the distance from the nearest data, as indicated by the decreasing local maxima and the increasing local minima.

### 2.3 Evaluation of Predicted Nucleosome Position Accuracy

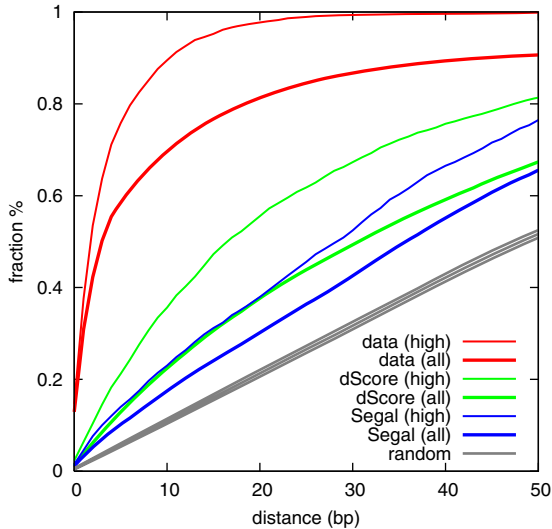
We have previously created a set of 50,814 estimated nucleosome dyad positions in yeast based on the experimental data of Field *et al.* [14]. The genomic positions of these dyads were estimated by applying a simple peak-detection algorithm to a nucleosome occupancy map, and a confidence score derived from the number



**Fig. 2.** *S. cerevisiae* chromosome II: Experimental data (blue stars) with a gap spanning the coding sequence for ribosomal protein RPL4A (approximately 300,000–301,400) Top: nucleosome core posterior probabilities inferred from experimental data using geometric linker (red) or uniform linker (green). Bottom: sequence-model scores (green) are added as additional evidence and nucleosome positions recomputed.

of overlapping reads was associated with each dyad [11]. In order to evaluate the positional accuracy of the predictions based on the two different sequence-model scores described above, we compare the predicted dyad positions (local maxima in the posterior probability of being in the dyad state) to the experimental benchmark set and compute the fraction of the positions in the experimental set that are within  $X$  nucleotides of a predicted dyad.

Posterior probabilities of nucleosome positions were computed using three different input tracks (one at a time): (a) the experimental Field occupancy map, (b) the dScores, and (c) the Segal raw binding scores. Predicted dyad positions were then compared to the entire benchmark set and to a small subset of the highest scoring positions (Fig 3). Because the estimated positions being used as the benchmark were derived from the same data used in (a), one would expect a near perfect concordance, and in fact the majority of the 50,814 dyads have corresponding predictions within 3 bp. The fact that the predictions based on the experimental dataset do not match up more exactly to the positions estimated using a simple peak-detection approach highlights the strengths of using a sequence model which simultaneously integrates all available information along the entire sequence. For example, if the experimental data indicates a sharply demarcated NFR, the edges of the NFR will affect the positioning of adjacent nucleosomes. These effects are automatically considered by the DBN but not by a simplistic



**Fig. 3.** Dyad positions inferred by the DBN using experimental data (red), dScores (green), or Segal raw binding scores (blue), compared to previously estimated dyad positions. Each pair of curves represents an evaluation over the entire set of 50,814 estimated dyad positions (all) and the top-scoring 3,180 (high). Each curve represents the fraction  $y$  of the estimated dyad positions for which a dyad was predicted by the DBN to within  $x$  nucleotides. The grey curves represents the performance that would be expected by chance (mean, and mean  $\pm$  one standard deviation, from simulations).

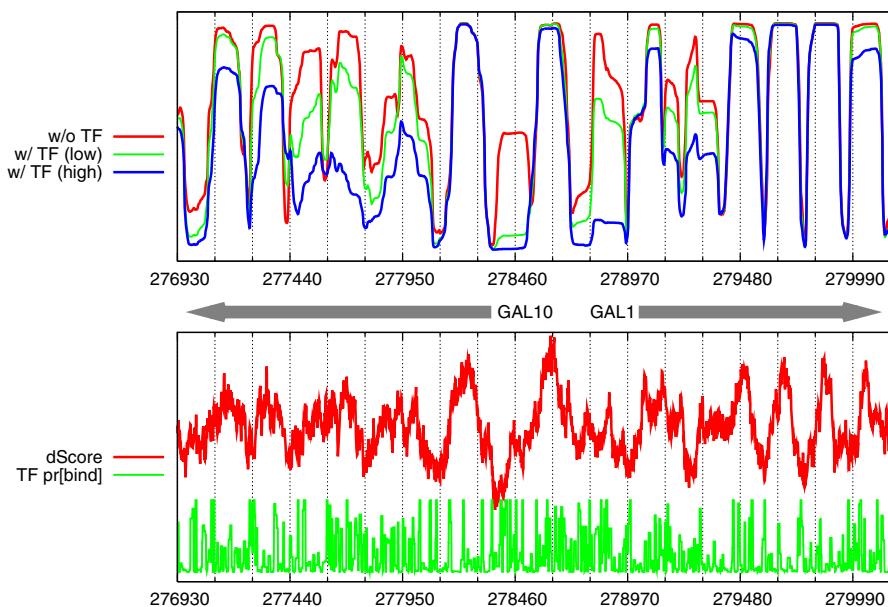


peak-detection approach. For the purposes of comparing to predictions based on sequence scores, this comparison to predictions based directly on the data provides an upper bound on the performance of any other method.

The dyad positions predicted using either type of sequence-based scores are both much less similar to the benchmark positions, although for both models the high-scoring benchmark dyads are more likely to be predicted accurately. At a maximum distance of 15 nucleotides between a benchmark dyad and a predicted dyad, corresponding to a 90% overlap between the reference nucleosome core and the prediction, the dScore-based predictions match 47% of the high-scoring subset and 31% of the entire set, compared to 31% and 24% respectively for the Segal-based predictions, and the 16% that would be expected by chance. All three sets of predictions contained very similar numbers of predicted dyads ( $\sim 62,500$ ), so these accuracy figures are directly comparable.

## 2.4 Competition with Transcription Factors

Histone proteins do not interact with the DNA to form nucleosomes in isolation, but rather compete dynamically with other DNA binding factors. To illustrate how this notion of competition can be incorporated into our model, we show an example of combining nucleosome-sequence scores with a landscape of transcription factor binding probabilities. We scanned the yeast genome using the



**Fig. 4.** *S. cerevisiae* chromosome II: competition with transcription factors destabilizes weakly positioned nucleosomes first. Top: nucleosome positions inferred from dScores without (red) and with low (green) and high (blue) levels of TF competition. Bottom: dScores (red) and TF binding probability landscape (green).

112 DNA-binding protein sequence specificities described by Badis *et al.* [15], and created an overall TF landscape by taking the maximum resulting binding probability at each position (see *Methods* for details). This information was then used in parallel with the dScores described earlier, and results are shown in Figure 4. This region of yeast chromosome II has two genes transcribed in opposite directions, with transcription start sites separated by approximately 600 bp. Immediately upstream of each TSS is a region of very high AT-content which includes strong matches for several TFs including SIG1 and PHO2. The figure shows that including the TF binding landscape almost completely eliminates the formerly weakly predicted nucleosome upstream of the GAL10 TSS while not significantly affecting the most strongly predicted nucleosomes.

### 3 Discussion

We have developed a novel solution to the problem of predicting nucleosome positions along a chromosome by incorporating arbitrary sources of information within a coherent probabilistic framework. Previous approaches have solved only part of this problem, using either sequence information alone or experimental data alone. Using sequence-based evidence in combination with experimental data provides a mechanism for interpreting the experimental data while filling in gaps using sequence predictions. Gaps in experimental data can be a significant problem in organisms with much larger (and more highly repetitive) genomes than yeast, where even genome-wide assays of nucleosome positioning produce relatively sparse data sets [16,17]. Combining multiple input tracks also permits us to investigate the relative impacts of different factors on the nucleosome landscape. Two different sequence-models could even be combined to see whether, jointly, they can make more accurate predictions than either one individually.

While we acknowledge the ongoing debate as to the impact *in vivo* of sequence-directed nucleosome positioning, we believe that predictive models that can incorporate the mechanisms that affect nucleosome positioning will increase our understanding of the chromatin structure and the impact it has on gene regulation and expression. Based on our genome-wide comparison of nucleosome positions estimated from an *in vivo* dataset to those predicted using dScore, we find that roughly 15% more of the nucleosome cores are predicted with at least a 90% overlap than would be expected by chance. The remaining nucleosomes are likely to follow a statistical positioning pattern, which this DBN naturally models. It may be interesting to explicitly compare a nucleosome-occupancy probability computed using purely local information to the probability computed by a full sequential model in order to understand which nucleosomes are predicted to be well-positioned due to a locally strong sequence signal and which might be predicted to be well-positioned as a result of a nearby, strongly-positioned “barrier” [10].

In this study, we opted not to evaluate our methods by computing a correlation between the posteriors produced by our model and an experimentally determined nucleosome occupancy profile [1]. Empirically, such profiles generally

exhibit a strong dependence on local GC-content; consequently, a simple sliding window of GC-content yields a pseudo nucleosome positioning signal that correlates at 0.70 with an empirical *in vitro* profile and between 0.56 and 0.63 for three *in vivo* sets from [12]. Although the inherent GC-richness of the nucleosome cores and AT-richness of the linkers will naturally produce this type of correlation, our concern is that the known GC-bias of the Illumina high-throughput sequencing will further enhance this effect. In contrast, a separate *in vivo* data set [14], from the same lab but based on the Roche 454 sequencing platform, has a lower correlation with local GC-content ( $r=0.42$ ), which is consistent with the lower GC-bias previously observed with these longer reads [18]. A recent study investigating the impact of chromatin structures on laboratory DNA manipulation [19] also noted that the sequencing bias toward higher read-density in GC-rich regions of Illumina-based deep sequencing [20] can result in a misleading overrepresentation of sequence reads in GC-rich DNA that will correlate strongly with GC-rich genomic features. The dScore was explicitly designed to be insensitive to GC-content across its analysis window (301 bp), and is less correlated ( $r = 0.46$ ) with GC-content computed on a smaller scale (71 bp) than the Segal raw binding score ( $r = 0.74$ ). Rather than trying to reproduce the wandering baseline seen in experimental nucleosome occupancy maps, we choose to focus on trying to accurately predict the most likely positions of linkers *vs* cores. In the posterior probabilities produced by our model, a deep null indicates a highly confident linker position and in turn a highly confident adjacent nucleosome, while regions of greater uncertainty are characterized by smaller differences between adjacent local maxima and local minima.

We believe that our discriminative framework for incorporating arbitrary heterogeneous scores directly into a sequential model will also prove useful in other segmentation applications in which a score can be interpreted directly as a label probability and may not lend itself well to being modeled using Gaussian mixtures in a generative framework—one possible example being inferring copy number variation from experimental data [21]. This framework can also be extended by using indicator variables [22] to explicitly allow for missing data or to specify, for example, that when two input tracks are both present only one of the two should be used.

## 4 Methods

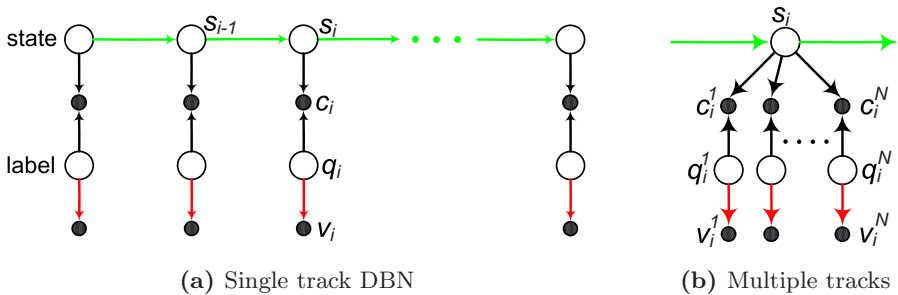
### 4.1 A Dynamic Bayesian Network for Nucleosome Prediction

The DBN that we use in this work is similar to a previous DBN-based method we developed to predict transmembrane protein topology from sequence [23], and is implemented using the Graphical Models Toolkit (GMTK) [24]. The task addressed by *Philius*, the topology prediction DBN, is the segmentation of an input protein into a series of non-overlapping regions belonging to one of three classes: *membrane*, *inside*, or *outside*. In this nucleosome prediction task, our goal is even simpler because there are only two classes of interest: *nucleosome core* and *linker*. Philius introduced a novel approach to using partially labeled data during

training which we will further generalize here. Typically, when labeled data is used to train an HMM (*i.e.* supervised training), the label accompanying each observation (*e.g.* nucleotide or amino acid) specifies the value of the associated “hidden state”. Philius allows for a more flexible relationship between the label and the state during training: a one-to-many relationship is defined between the labels and the states, and a special “wildcard” label allows the state variable to take on *any* value that is otherwise consistent with the topology of the model. In the case of Philius, the wildcard label is used to address the uncertainty inherent in the segment boundaries—at each segment boundary, some labels were replaced by the wildcard in order to allow the model to make small adjustments to the boundary locations during training. For the purposes of nucleosome prediction, we exploit this idea to define a similarly flexible relationship between labels and states, although in the model presented here, the labels are not observed in the traditional sense—instead they are constrained by the evidence.

Philius uses a two-pass decoding process that makes use of so-called “soft” labels to find the protein topology that maximizes the posterior probabilities at each position while obeying the grammar constraints required by the membrane topology. In this work, we show that a similar mechanism can be used to incorporate a variety of information sources to predict nucleosome positioning while obeying the grammar constraints required by the chromatin “topology”.

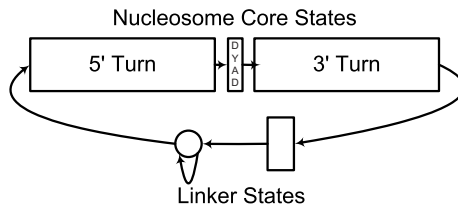
Figure 5a shows the graphical model of our DBN, in which a single track of virtual evidence is incorporated as a soft constraint on the value of the *label* node. For simplicity, this graphical model omits the portion of the graph which takes care of the counting for the fixed-duration states. This counting mechanism is implemented exactly as in Philius [23]. To fully define the nucleosome positioning DBN, in addition to the graphical model shown in Figure 5a, the precise form of the relationship between each node and its parent(s) must be defined. We will proceed by describing each of the DBN components in turn, starting with the Markov chain over states, then the relationship between each connected state and label pair  $(s_i, q_i)$  joined by the observed child  $c_i$ , and finally how the input



**Fig. 5.** Graphical models for (a) the DBN with a single track of evidence and (b) a single frame showing the incorporation of an arbitrary number  $N$  of evidence tracks. The small black nodes represent the virtual evidence, the white nodes represent hidden variables, the subscript  $i$  refers to the genomic position and the superscripts in (b) index the evidence tracks.

model scores, experimental data, or other types of information are injected into the DBN via the relationship between the label node  $q_i$  and the virtual evidence node  $v_i$ .

**A Markov Chain over Hidden States.** Our model consists of five states: three to model the fixed-length nucleosome and two to model the variable-length linker. The three states that are used to describe the nucleosome and their associated lengths are the *dyad* (5 bp), and the 5' and 3' *turns* (71 bp each), where the dyad refers to the central position of the nucleosomal sequence, at the axis of symmetry of the histone core. The linker is described using two states: a fixed-length state (9 bp), and a state with a geometrically-distributed length (implemented as a simple self-looping state, with minimum length 1 bp). Together these two states capture the steric hindrance constraint between adjacent nucleosomes, enforcing a minimum linker length of 10 bp, while also allowing for arbitrarily long linkers. The state transition diagram is shown in Figure 6 and consists of a simple cycle in which each state has only one possible *next* state, meaning that when a *change* in state is to occur, there is only one possible new, different state given the current state. This simple sequence of states defines the nucleosome “grammar”. For simplicity, the initial state is always defined to be the geometric-length linker state. This hard constraint greatly reduces the complexity of the inference while having relatively little effect on the predictions. For all subsequent states, the conditional relationship between each state and the previous state  $Pr[s_i|s_{i-1}]$  is defined according to the deterministic grammar described above, with the exception of the self-looping linker state which transitions to the next state (the 5' turn) with probability  $p$  or remains in the linker state with probability  $1 - p$ . The duration model realized by this self-looping state is a geometric distribution,  $Pr[k] = (1 - p)^{k-1}p$  for  $k > 0$ , with mean  $1/p$ . By using a feature in GMTK that allows for exponential weights to be applied to any edge in the DBN, we can also run our model with a completely unbiased linker model. We do this by setting a weight of 0 on the state-transition edge: this exponential weight is applied to any non-zero probability in the state-transition matrix, causing all non-zero values in the matrix to become 1. In this mode, the  $Pr[k]$  defined above is equal to 1 for all values of  $k$ . The effect of this exponential weight is similar to that of the temperature constant in a Boltzmann model,



**Fig. 6.** State transition diagram. The width of each rectangular state is proportional to the duration specified for that state. The circular state represents the self-looping linker state which follows a geometrical duration distribution.

albeit inverted: a weight of 0 corresponds to an infinite temperature at which all possible outcomes become equally likely.

**Virtual Evidence Constraints.** While the backbone of our model is the same Markov chain over hidden states that exists in the traditional HMM, the relationship between the hidden state and the “observation” is quite different. While each state in an HMM is traditionally thought of as “emitting” a particular discrete or continuous observed random variable, and the probability distribution over the observed variable is conditioned on the hidden state, our model has a more discriminative flavor in which the information available at each genomic position is used to directly influence the local probability distribution over possible state assignments. The result is that the probability of a particular sequence of state assignments is weighted according to the information available at each position. This direct influence on the local probability over the possible assignments to the state variable is accomplished using the concept of “virtual evidence” [23,25,26], as will be described in more detail below. Below each state in the graphical model, a typical HMM would have a single observed node  $o_i$ , dependent on the parent state  $s_i$  according to some distribution  $Pr[o_i|s_i]$ . In this DBN, we have instead two distinct relationships: the first is a deterministic relationship between the state  $s_i$ , the label  $q_i$ , and the virtual evidence node  $c_i$ :  $Pr[c_i|s_i, q_i]$ . This construct, in which  $c_i$  is called an *observed child* because it induces a relationship between its parents, is used to define which states are consistent with a particular label:  $c_i$  is observed to be equal to 1, and the table  $Pr[c_i = 1|s_i, q_i]$  implements an indicator function  $I(s_i, q_i)$ , which is equal to 1 if  $s_i$  and  $q_i$  are consistent with one another, and otherwise is equal to 0.

The second probabilistic relationship shown in the graphical model is between the label  $q_i$  and a second virtual evidence node  $v_i$ , and is defined as  $Pr_i[v_i = 1|q_i]$ . We add the subscript  $i$  to this conditional relationship to indicate that it depends on the current position,  $i$ , unlike the relationship between the *state* and the *label*, and unlike the observation distribution in a typical time-homogeneous HMM. Finally, we assign a uniform marginal distribution over the possible values of  $q_i$ :  $Pr[q_i = Q] = 1/|Q|$  where  $Q$  represents a specific label, and  $|Q|$  is the cardinality of the discrete label variable.

**Joint Probability Distribution.** We can now give the equation for the probability of a particular assignment to all of the hidden nodes, in other words to a particular sequence of states  $\mathbf{s}$ , and a particular sequence of labels  $\mathbf{q}$ :

$$Pr[\mathbf{s}, \mathbf{q}] \propto \left( Pr[s_1] \prod_{i=2}^N Pr[s_i|s_{i-1}] \right) \left( \prod_{i=1}^N \mathbf{I}[s_i, q_i] Pr[q_i] Pr_i[v_i|q_i] \right)$$

in which we use the indicator function  $I(s_i, q_i)$  in place of  $Pr[c_i = 1|s_i, q_i]$ . The indicator function  $\mathbf{I}[s_i, q_i]$  will cause all inconsistent pairs of sequences  $\mathbf{s}$  and  $\mathbf{q}$  to have probability zero. Considering only the subset of sequence pairs that are self-consistent  $\{\bar{\mathbf{s}}, \bar{\mathbf{q}}\}$ , this probability can be restated as:

$$Pr[\bar{s}, \bar{q}] \propto \left( Pr[\bar{s}_1] \prod_{i=2}^N Pr[\bar{s}_i | \bar{s}_{i-1}] \right) \left( \prod_{i=1}^N Pr[\bar{q}_i] Pr_i[v_i | \bar{q}_i] \right)$$

in which the first term in parentheses scores the sequence of states and enforces the grammar defined by the state-transition matrix, while the second term incorporates the virtual evidence at each position. Finally, we sum over all consistent label sequences  $\bar{q}$ , to find the probability of a particular sequence of states:

$$Pr[\bar{s}] \propto \left( Pr[\bar{s}_1] \prod_{i=2}^N Pr[\bar{s}_i | \bar{s}_{i-1}] \right) \left( \sum_{\bar{q}} \prod_{i=1}^N Pr[\bar{q}_i] Pr_i[v_i | \bar{q}_i] \right)$$

This probability can be computed efficiently using the junction tree algorithm, which is a generalization of the forward-backward algorithm for HMMs, because of the underlying tree structure of the graph. We can similarly compute the posterior probabilities for the state variable at each position, and this will be the standard output of our model—specifically we plot the posterior  $Pr_i[core]$  computed by summing the posterior probabilities of the three nucleosome states (the dyad and the 5’ and 3’ turns). Furthermore, multiple tracks of evidence can be incorporated into the model simply by replicating the evidence portion of the model as shown in Figure 5b. All of the information available at each genomic position will be used to infer the probabilities of the possible assignments to the state variable at that position.

**Evidence Track Definition.** We have defined the state space of our model but we have not yet precisely defined either the labels or the virtual evidence that we intend to use to define the function  $Pr_i[v_i | q_i]$ . We describe three possible sources of information to be used as inputs to our model, although our intent here is to describe a framework in which arbitrary sources of information can be combined in a principled manner to predict nucleosome positioning along a chromosome. The three types of nucleosome-positioning information that we describe are: a) scores from a DNA-sequence model of nucleosome positioning; b) nucleosome-occupancy data from a high-throughput sequencing experiment; and c) a transcription factor “landscape”. The first two types of information can each be used as the sole source information, while the TF landscape is shown used in conjunction with scores from a sequence model. The one-to-many relationship between each label variable and the associated state variable is customized for each type of input data.

*Sequence model scores.* Assuming that a sequence model score  $z_i$  can be interpreted as a log-ratio, in other words a choice between two hypotheses, we define  $q_i$  to be a binary label such that  $q_i = 1$  corresponds to the dyad state, and  $q_i = 0$  corresponds to any non-dyad state. The virtual evidence node,  $v_i$  is also a binary random variable, although we always observe  $v_i = 1$  for all  $i$ . We assign uniform marginal probability distributions to both of these binary variables, and then use the law of total probability to find that the sum of the conditional probabilities  $Pr[v_i = 1 | q_i = 1]$  and  $Pr[v_i = 1 | q_i = 0]$  is equal to 1. Furthermore, we define



the log-ratio of these two conditional probabilities to equal the aforementioned score,  $z_i$ , and therefore:

$$Pr[v_i = 1|q_i = 1] = \frac{1}{1 + e^{-z_i}} \quad \text{and} \quad Pr[v_i = 1|q_i = 0] = \frac{1}{1 + e^{z_i}}$$

*Experimental data.* Experimental data derived from a microarray or sequencing assay can similarly be interpreted as a log-ratio and supplied as an evidence track exactly as described for the sequence scores above.

*Transcription factor binding probabilities.* The third type of input information that we consider is a binding probability track representing one or more TFs. We model the relative affinity of a binding site to a particular transcription factor  $X$  using a position weight matrix (PWM) as described in [27]. Assuming that a TF can only bind in the absence of a nucleosome, *i.e.* in a linker region, we define  $q_i$  such that  $q_i = 1$  corresponds to either linker state, and  $q_i = 0$  corresponds to *any* state. A high TF-binding probability (high probability that  $q_i = 1$ ) will therefore result in a higher probability of being in a linker state, while a low TF-binding probability (high probability that  $q_i = 0$ ) will have little to no effect.

## References

1. Lubliner, S., Segal, E.: Modeling interactions between adjacent nucleosomes improves genome-wide predictions of nucleosome occupancy. *Bioinformatics* 25, 1348–1355 (2009)
2. Yuan, G.C., Liu, J.S.: Genomic Sequence is Highly Predictive of Local Nucleosome Depletion. *PLoS Comp. Biol.* 4, e13 (2008)
3. Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thøaström, A., Field, Y., Moore, I.K., Wang, J.Z., Widom, J.: A genomic code for nucleosome positioning. *Nature* 44, 772–778 (2006)
4. Peckham, H.E., Thurman, R.E., Fu, Y., Stamatoyannopoulos, J.A., Noble, W.S., Struhl, K., Weng, Z.: Nucleosome positioning signals in genomic DNA. *Genome Research* 17, 1170–1177 (2007)
5. Wasson, T., Hartemink, A.J.: An ensemble model of competitive multi-factor binding of the genome. *Genome Research* 19, 2101–2112 (2009)
6. Yuan, G.C., Liu, Y.J., Dion, M.F., Slack, M.D., Wu, L.F., Altschuler, S.J., Rando, O.J.: Genome-scale identification of nucleosome positions in *S. cerevisiae*. *Science* 309, 626–630 (2005)
7. Lee, W., Tillo, D., Bray, N., Morse, R.H., Davis, R.W., Hughes, T.R., Nislow, C.: A high-resolution atlas of nucleosome occupancy in yeast. *Nature Genetics* 39, 1235–1244 (2007)
8. Yassour, M., Kaplan, T., Jaimovich, A., Friedman, N.: Nucleosome positioning from tiling microarray data. *Bioinformatics* 24, i139–i146 (2008)
9. Bilmes, J., Bartels, C.: Graphical Model Architectures for Speech Recognition. *IEEE Signal Processing Magazine* 22, 89–100 (2005)
10. Mavrich, T.N., Ioshikhes, I.P., Venters, B.J., Jiang, C., Tomsho, L.P., Qi, J., Schuster, S.C., Albert, I., Pugh, B.F.: A barrier nucleosome model for statistical positioning of nucleosomes throughout the yeast genome. *Genome Research* 18, 1073–1083 (2008)



11. Reynolds, S.M., Bilmes, J.A., Noble, W.S.: Learning a weighted sequence model of the nucleosome core and linker yields more accurate predictions in *Saccharomyces cerevisiae* and *Homo sapiens* (in submission)
12. Kaplan, N., Moore, I.K., Fondufe-Mittendorf, Y., Gossett, A.J., Tillo, D., Field, Y., LeProust, E.M., Hughes, T.R., Lieb, J.D., Widom, J., Segal, E.: The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature* 548, 362–366 (2009)
13. Sun, W., Xie, W., Xu, F., Grunstein, M., Li, K.-C.: Dissecting Nucleosome Free Regions by a Segmental Semi-Markov Model. *PLoS One* 4, e4721 (2009)
14. Field, Y., Kaplan, N., Fondufe-Mittendorf, Y., Moore, I.K., Sharon, E., Lubling, Y., Widom, J., Segal, E.: Distinct Modes of Regulation by Chromatin Encoded through Nucleosome Positioning Signals. *PLoS Comp. Biol.* 4, e1000216 (2008)
15. Badis, G., Chan, E.T., van Bakel, H., Pena-Castillo, L., Tillo, D., Tsui, K., Carlson, C.D., Gossett, A.J., Hasinoff, M.J., Warren, C.L., Gebbia, M., Talukder, S., Yang, A., Mnaimneh, S., Terterov, D., Coburn, D., Yeo, A.L., Yeo, Z.X., Clarke, N.D., Lieb, J.D., Ansari, A.Z., Nislow, C., Hughes, T.R.: A library of yeast transcription factor motifs reveals a widespread function for Rsc3 in targeting nucleosome exclusion at promoters. *Mol. Cell* 32, 878–887 (2008)
16. Barski, A., Cuddapah, S., Cui, K., Roh, T.Y., Schones, D.E., Wang, Z., Wei, G., Chepelev, I., Zhao, K.: High-resolution profiling of histone methylations in the human genome. *Cell* 129, 823–837 (2007)
17. Schones, D.E., Cui, K., Cuddapah, S., Roh, T.Y., Barski, A., Wang, Z., Wei, G., Zhao, K.: Dynamic regulation of nucleosome positioning in the human genome. *Cell* 132, 887–898 (2008)
18. Harismendy, O., Ng, P.C., Strausberg, R.L., Wang, X., Stockwell, T.B., Beeson, K.Y., Schork, N.J., Murray, S.S., Topol, E.J., Levy, S., Frazer, K.A.: Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol.* 10, R32 (2009)
19. Teytelman, L., Özyaydin, B., Zill, O., Lefrançois, P., Snyder, M., Rine, J., Eisen, M.B.: Impact of Chromatin Structures on DNA Processing for Genomic Analyses. *PLoS One* 4, e6700 (2009)
20. Dohm, J.C., Lottaz, C., Borodina, T., Himmelbauer, H.: Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research* 36, e105 (2008)
21. Marioni, J.C., Thorne, N.P., Tavaré, S.: BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data. *Bioinformatics* 22, 1144–1146 (2006)
22. Hoffman, M.M., Buske, O.J., Bilmes, J.A., Noble, W.S.: Segway: a dynamic Bayesian network method for segmenting genomic data (in preparation)
23. Reynolds, S.M., Käll, L., Riffle, M.E., Bilmes, J.A., Noble, W.S.: Transmembrane topology and signal peptide prediction using dynamic Bayesian networks. *PLoS Comp. Biol.* 4, e1000213 (2008)
24. Bilmes, J., Zweig, G.: The Graphical Models Toolkit: An Open Source Software System for Speech and Time-Series Processing. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE Press, New York (2002)
25. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco (1988)
26. Reynolds, S.M., Bilmes, J.A.: Part-of-speech tagging using virtual evidence and negative training. In: *Proc. HLT and EMNLP*, pp. 459–466. IEEE Press, New York (2005)
27. Granek, J.A., Clarke, N.D.: Explicit equilibrium modeling of transcription-factor binding and gene regulation. *Genome Biol.* 6, R87 (2005)

# Dense Subgraphs with Restrictions and Applications to Gene Annotation Graphs

Barna Saha<sup>1</sup>, Allison Hoch<sup>2</sup>, Samir Khuller<sup>3</sup>, Louiqa Raschid<sup>4</sup>,  
and Xiao-Ning Zhang<sup>5</sup>

<sup>1</sup> Research supported by NSF Award CCF-0728839  
Department of Computer Science, University of Maryland,  
College Park, MD 20742

[barna@cs.umd.edu](mailto:barna@cs.umd.edu)

<sup>2</sup> Research supported by NSF REU Supplement to Award CCF-0728839  
Department of Computer Science, University of Maryland, College Park, MD 20742

[allie@umd.edu](mailto:allie@umd.edu)

<sup>3</sup> Research supported by NSF Award CCF-0728839 and a Google Research Award  
Department of Computer Science and UMIACS, University of Maryland,  
College Park, MD 20742

[samir@cs.umd.edu](mailto:samir@cs.umd.edu)

<sup>4</sup> Research supported by NSF Award IIS-0430915 and IIS-0960963  
UMIACS and Robert H. Smith School of Business, University of Maryland,  
College Park, MD 20742

[louiqa@umiacs.umd.edu](mailto:louiqa@umiacs.umd.edu)

<sup>5</sup> Research supported by Department of Biology, St. Bonaventure University,  
St. Bonaventure, NY 14778 and Department of Cell Biology and Molecular Genetics,  
University of Maryland, College Park, MD 20742

[xzhang@sbu.edu](mailto:xzhang@sbu.edu)

**Abstract.** In this paper, we focus on finding complex annotation patterns representing novel and interesting hypotheses from gene annotation data. We define a generalization of the densest subgraph problem by adding an additional distance restriction (defined by a separate metric) to the nodes of the subgraph. We show that while this generalization makes the problem NP-hard for arbitrary metrics, when the metric comes from the distance metric of a tree, or an interval graph, the problem can be solved optimally in polynomial time. We also show that the densest subgraph problem with a specified subset of vertices that have to be included in the solution can be solved optimally in polynomial time. In addition, we consider other extensions when not just one solution needs to be found, but we wish to list all subgraphs of almost maximum density as well. We apply this method to a dataset of genes and their annotations obtained from The Arabidopsis Information Resource (TAIR). A user evaluation confirms that the patterns found in the distance restricted densest subgraph for a dataset of photomorphogenesis genes are indeed validated in the literature; a control dataset validates that these are not random patterns. Interestingly, the complex annotation patterns potentially lead to new and as yet unknown hypotheses. We perform experiments to determine the properties of the dense subgraphs, as we vary parameters, including the number of genes and the distance.

## 1 Introduction

Biological knowledge is increasingly being represented using graphs, e.g., protein interactions, metabolic pathways, gene regulation, gene annotation, etc. Finding highly dense regions in graphs is a problem of both theoretical [17,12,3,14] and practical importance. *Density* is one quantitative measure of the connectedness of a subgraph and is defined as the ratio of the number of induced edges to the number of vertices in the subgraph. Even though there are an exponential number of subgraphs, a subgraph of maximum density can be found in polynomial time [17,12,3]. In contrast, the *maximum clique* problem to find the subgraph of largest size having all possible edges is *NP*-hard; it is even *NP* hard to obtain any non-trivial approximation. Finding densest subgraphs with additional size constraints is *NP* hard [14]; yet, they are more amenable to approximation than the maximum clique problem. Moreover detecting only cliques can be somewhat restrictive, since interesting subgraphs missing a few edges are omitted by any such procedure.

In this paper, we apply the densest subgraph problem to the task of finding complex patterns in a *gene annotation graph* representing annotations of genes using terms from controlled vocabularies (CVs) or ontologies. We attempt to increase the biological meaning of subgraphs by favoring the inclusion of pairs of nodes that have a meaningful relationship within the ontology structure that was used to create the gene annotation graph; we do this by defining a distance metric  $d_H$  between pairs of nodes. The goal is to return dense subgraphs with vertices within the subgraph satisfying a distance threshold.

We introduce a new variant of densest subgraph problems in this paper, namely the *distance restricted* densest subgraph problem to capture this property. We are given a graph  $G = (V, E)$  as well as a distance metric  $d_H$  defined over pairs of vertices  $u, v \in V$ . The goal is to return a maximum density subgraph  $S \subseteq V(G)$ , such that in  $S$ , any pair of vertices are within distance  $\tau$  according to  $d_H$ .

Further, researchers may be interested in obtaining patterns containing pre-specified nodes. We refer to this as the *subset maximum density problem*, and this is described in Section 3. Finding only *one* dense subgraph may not suffice since the researchers may wish to find many complex annotation patterns. Thus, we address the problem of *all* maximum and *nearly* maximum dense subgraphs with distance/ subset restrictions in Section 4. We are the first to introduce and study the problem of detecting distance and subset restricted densest subgraphs.

In computational biology, there has been a body of work closely related to detecting dense subgraphs. Most of these papers concentrate on protein-protein interaction networks, where the goal is to cluster the network to detect densely connected molecular modules [30,15,19,24], that can possibly identify protein families and molecular complexes [41], or even identify missing interactions [32] and annotations [21]. Work by Newman [22] studies community detection in metabolic and regulatory networks. Communities are characterized by high intra and sparse inter connectivity. Many of these works on community detection can benefit by application of distance restricted dense subgraphs problem and its extensions.

The works of [119] consider *clustering coefficients* for a measure of density on the neighborhood of each node. It is defined as the ratio of edges among the neighbors to the maximum possible number of edges. Thus an alternative measure for density can be to compute clustering coefficient of the entire subgraph. However it tends to find very small subgraphs and is not effective.

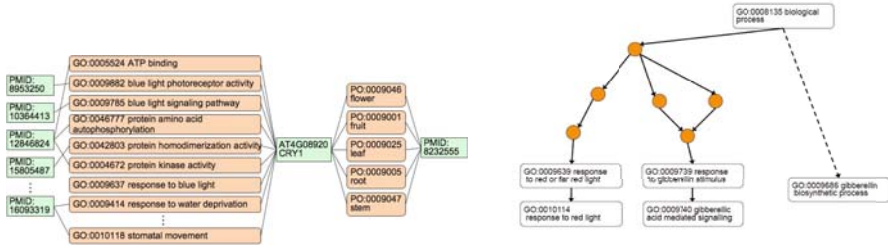
## 1.1 Gene Annotation Data

Knowledge about genes has been captured in publicly available bibliographic resources such as PubMed [276] and PubMed Central [286], general purpose resources such as Entrez Gene [520], and in more focused model organisms or domain specific collections such as The Arabidopsis Information Resource (TAIR) [8169]. In order to improve interoperability, various communities have created a number of ontologies such as the Gene Ontology (GO) [711], the Plant Ontology (PO) [26], and the Unified Medical Language System (UMLS) [231]. Data entries (records) in a resource are typically annotated with concepts or controlled vocabulary (CV) terms from one or more of these ontologies, creating a rich Web of annotation knowledge.

We focus on The Arabidopsis Information Resource (TAIR) [8169]. A scientist can typically visit a page that provides a rich synopsis of a TAIR gene and then follow links to reach genotype and phenotype annotation data, publications, organism specific data, ESTs, pathway data, etc. Annotations in TAIR are associated with explanations or evidence codes reflecting the underlying methodology supporting the annotation. While TAIR is a valued and much visited portal that fuels the progress of scientific research, it also requires that scientists spend many hours manually clicking through web pages and following links, to create a subset of annotation knowledge for pattern discovery. Scientists often use simple tools such as a spreadsheet to maintain this subset of annotation knowledge. Increasingly, there is a need for more sophisticated tools to help the scientist integrate, analyze and visualize this knowledge.

We illustrate this using an example tool for integrating TAIR annotation data. The *LSLink* system [18] can be used to specify a protocol to create a background *LSLink* dataset of hyperlinked data records and their annotations. The protocol follows hyperlinks from each TAIR gene, and integrates the corresponding GO annotations, PO annotations, and the publications in PubMed that support the annotation. Some sample output of the integration protocol and the *LSLink* dataset is illustrated in Fig. 1(a) where we visualize the annotations for gene *CRY1*. The GO annotations are on the left side and the PO annotations on the right. Each includes the identifier and the label for the Controlled Vocabulary (CV) term. In addition, the figure includes the PubMed publications that support the annotations. As of January 2009, there were 17 GO annotations and 5 PO annotations for *CRY1*. The figure illustrates only some of the annotations (due to lack of space).

The *LSLink* annotation dataset of Fig. 1(a) represents knowledge culled from multiple research projects and their accompanying publications. The challenge for the scientist is to mine these datasets to discover important patterns.



(a) Associations between GO and PO CV Terms Gene CRY1 (b) Semantic relationships in Gene Ontology GO

Fig. 1. TAIR

Consider the gene *GA30X1* and a simple pattern of a pair comprising the GO CV term *gibberellic acid mediated signaling* and the PO CV term *germination*; it is meaningful since *GA30X1* regulates seedling growth. While these simple patterns are somewhat interesting, in order to capture biological knowledge, the scientist would be interested in finding a more complex pattern. Identifying a complex pattern in the annotations of a single gene may be non-trivial for a gene such as *CRY1* which has many annotations since the scientist has to consider many pairs of annotations and many groups of CV terms. However, the real challenge is even more difficult. While a pattern comprising a group of GO and PO terms annotating a single gene may correspond to a meaningful biological phenomenon, it may not be an interesting discovery. This is because it is annotating a single gene and the knowledge may be well known. A truly interesting discovery of knowledge that is as yet unknown, typically would require that the scientist solve the greater challenge of finding a pattern of a group of PO terms and GO terms that annotated *multiple genes*. Identifying such a co-occurrence pattern for a group of as yet unrelated genes can lead to the *gold standard* of an interesting discovery that would lead to actionable hypothesis, e.g., an experiment to verify the pattern.

The second challenge is that the GO and PO terms that form a pattern are not independent but they occur within a (hierarchical) ontology structure. Controlled vocabulary (CV) terms that are closer to each other in the hierarchy may be more closely related in meaning. Consider the fragment of the GO hierarchy of Fig. 1(b). This fragment illustrates some of the GO terms that annotate the TAIR gene *GA30X1*. The labeled rectangular nodes annotate the gene while the circular nodes are placeholder GO CV terms in the ontology that do not annotate *GA30X1*. We note that the following 2 terms, *response to gibberellin stimulus* and *gibberellic acid mediated signaling*, are more closely related whereas the pair of terms, *response to red light* and *gibberellin biosynthetic process* may appear to be unrelated. A complex pattern that included the first pair is more likely to be meaningful in comparison to a complex pattern that included the second pair. Two nodes in the ontology graph that have a smaller shortest path distance are assumed to be more closely related and therefore more biologically meaningful, compared to a pair that are farther apart in the structure.

## 1.2 Gene Annotation Graph and Notion of Density

We can formalize our problem as follows: We are given two ontologies, GO and PO and a collection of genes  $\mathcal{G}$ , that are associated with some subsets of the CV terms in the two ontologies. In other words, each gene is annotated (associated) with a set of GO and PO nodes as seen in Fig. 1(a). We can represent this data in the form of a bipartite graph  $G = (A, B, E)$  between the set of GO nodes and the set of PO nodes. The bipartite graph is a weighted graph where each edge is labeled with a set of gene names, such that each gene is annotated with the corresponding GO and PO nodes.

Each CV term in the GO (or PO) ontology has a vertex representing it in  $A$  (or  $B$ ). If there are  $t$  genes  $g_1, g_2, \dots, g_t \in \mathcal{G}$  containing the CV terms corresponding to vertices  $u \in A$  and  $v \in B$  in their annotations, then an edge is added between  $u$  and  $v$  in  $G$ , with weight  $w'(u, v) = t$ . We will often refer to this bipartite graph  $G$  as the annotation graph. We note that while we illustrate our algorithms using this GO PO bipartite graph, our algorithm works equally well for general graphs.

If the set of genes of interest are richly annotated with GO and PO terms, then the scientist has to examine a large annotation graph  $G$ . Even a simple yet meaningful visualization of the annotation graph is non trivial. Our high level objective is to discover complex patterns involving multiple genes that are co-annotated with the same subset of GO and PO terms. One way to do this is to identify large cliques in bipartite graphs. To be more flexible in finding interesting patterns, we instead look for densest subgraphs that find a large set of genes sharing a lot of common GO and PO terms; at the same time we would like the GO and PO terms to be closely related leading to the distance restriction.

Another formulation may consider the genes and their annotations by GO, PO nodes as a hypergraph. GO and PO nodes correspond to vertices as before, but now each gene is a hyperedge consisting of a set of GO and PO nodes. One related notion of density in a hypergraph is the ratio of hyperedges completely contained in a subgraph to the number of vertices present in that subgraph. Our algorithms for finding maximum density subgraphs work with hypergraphs as well. However this formulation may not be very useful in our context. A set of GO and PO nodes may be shared by a few genes; if there is a gene that in addition includes another GO or PO node that was not chosen, then it will not be included. The detection of this last gene might provide valuable information by discovering a missing annotation; but the hypergraph approach may not detect it.

We further consider two extensions to the problem that will be of interest to the scientist. Finding a single densest subgraph may not help the scientist explore all the interesting patterns in the annotation knowledge. One extension is to find all densest subgraphs. Further, there may be subgraphs that have density close to the maximum density that are also interesting, e.g., they include a different set of genes, or a different set of GO or PO terms, in comparison to the densest subgraph. Such diversity of subgraphs may also help the scientist discover interesting patterns. Thus, a natural generalization is to find all the

subgraphs of density close to the maximum. We refer to this as the *all almost maximum density subgraph problem*. Finally, scientists might be interested in filtering the densest subgraphs so that they contain a specified subset of GO or PO CV terms that are of interest to the scientist. We call this *subset maximum density problem*.

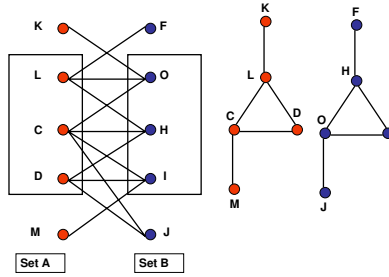
Our main contributions are as follows:

- In Section 2 we give a formal definition of the distance restricted dense subgraph problem. We show that for general metrics the problem is at least as hard to approximate as the well known independent set problem, and for special metrics such as trees and interval graphs it can be solved optimally in polynomial time. In addition, if we are willing to relax the distance threshold slightly we can solve it in polynomial time.
- In some cases there is a subset of GO and PO nodes that are to be studied, and we are specifically looking for subgraphs that contain these nodes. In Section 3 we show that the problem when a specified subset of vertices must be part of the subgraph can also be solved optimally in polynomial time as well.
- In Section 4 we show how Picard and Queyrannes’s framework [25] (developed to find a compact encoding of all  $s$ - $t$  min-cuts) can be adapted to find a collection of subgraphs whose density is close to the density of the maximum density subgraph. This framework can also be trivially extended for the generalizations we mentioned above (distance restricted subgraphs as well as the case when a subset of nodes must be part of the solution).
- Using a set of 10 photomorphogenesis genes and a set of 10 control genes, a user evaluation demonstrates that the *densest subgraph* for the photomorphogenesis genes returns many patterns that are validated by the literature. Further, the control genes validate that the results in the densest subgraph are not random patterns. Of more interest, we identified complex patterns of as yet not well known knowledge that could lead to new hypotheses. Results are reported in Section 5. We performed experiments on several other different set of genes and studied the properties of densest subgraphs and our algorithms on TAIR dataset. These additional results can be found in an extended version [29].

## 2 Distance Restricted Densest Subgraph Problem

In this section we are interested in the *distance restricted densest subgraph problem*. While our methods work for general graphs, in this framework we consider a bipartite graph  $G = (A, B, E)$  with two disjoint sets of vertices  $A$  and  $B$ , and a set of edges  $E$ . We are also given a distance function (say a metric)  $d_A$  ( $d_B$ ) that specifies distances between pairs of nodes in set  $A$  ( $B$ ). In addition, we are given distance thresholds  $\tau_A, \tau_B$ . The goal is to compute a densest subgraph  $G_S = (S_A, S_B, E_S)$  by choosing subsets  $S_A \subset A$  and  $S_B \subset B$  to maximize the density of the subgraph, which is defined as  $\frac{w'(E_S)}{|S_A|+|S_B|}$ . Here  $w'(E_S)$  denotes the weight of the edges in the subgraph induced by  $E_S$ . In addition, we require that





**Fig. 2.** An example of a bipartite graph  $G = (A, B, E)$  (left). GO and PO graphs are shown on the right.

for all pairs of vertices  $u, v \in S_A$  we have  $d_A(u, v) \leq \tau_A$ , and the same condition holds for pairs of vertices in  $S_B$ , namely that for all  $x, y \in S_B$  we have  $d_B(x, y) \leq \tau_B$ . Here  $G$  represents the annotation graph,  $A$  and  $B$  correspond to GO and PO nodes respectively. Distance function  $d_A$  ( $d_B$ ) comes from the shortest path metric of the GO (PO) ontology graph.

Consider the example in Fig. 2. We set  $\tau_A = \tau_B = 1$  (distance is defined by the shortest path metric). The densest subgraph that satisfies the distance constraints is as shown - formed by the nodes  $S_A = \{L, C, D\}$  and  $S_B = \{O, H, I\}$ . The number of edges is 7 in this induced graph giving a density of  $\frac{7}{6}$ . Note that the subgraph obtained by adding node  $J$  to  $G_S$  would have a higher density of  $\frac{9}{7}$ , but we cannot add  $J$  to the subgraph since  $d(H, J) > \tau_B$ . The proof of the following theorem is omitted for lack of space and can be found in the extended version [29].

**Theorem 1.** *When the distance function is an arbitrary metric, the problem is NP-hard and at least as hard to approximate as the maximum independent set problem [10].*

The relationship with the independent set problem explains why this problem is hard to approximate and it is not possible to develop approximation algorithms with good performance guarantee for this problem for general metrics. However, we next show that for many family of graphs this problem can be solved exactly. We identify a generic property of a metric, such that if the metric satisfies this property then we can solve the problem optimally in polynomial time.

### 2.1 Polynomial Time Algorithms for the Distance Restricted Densest Subgraph Problem

Let  $G = (A \cup B, E)$  and  $d_A$  and  $d_B$  be the two metrics. Let  $S_A \subseteq A$  and  $S_B \subseteq B$  form a densest subgraph in  $G$  such that any two vertices  $u, v \in S_A$ ,  $d_A(u, v) \leq \tau_A$ , for a given value  $\tau_A$  (and similarly for  $x, y \in S_B$ ,  $d_B(x, y) \leq \tau_B$ ).

For our specific problem, we encode the distance function between pairs of nodes in  $A$  and  $B$  by the shortest path distance in a given graph  $H$ .  $H$  will



have two components, one for  $A$  and one for  $B$ . Let  $n$  be the number of nodes in  $G$ . The high level idea is as follows: we wish to select a polynomial collection of subgraphs  $G_i = (A_i, B_i, E_i)$  for  $i = 1 \dots p(n)$ , and for each  $G_i$  compute the densest subgraph. The computed  $G_i$ 's satisfy the following two properties:

- **Distance property:** All pairs of nodes in  $A_i$  ( $B_i$ ) satisfy the pairwise distance constraint.
- **Subset property:** There exists some  $G_i$  that contains the true optimum solution  $G_S$ .

Since we find the densest subgraph within each  $G_i$ , we are guaranteed to find  $G_S$ . Thus a polynomial time algorithm for extracting  $G_i$ 's give a sufficient condition for the existence of a polynomial time algorithm for the *distance restricted densest subgraph* problem on  $G$ . To obtain the densest subgraph within each  $G_i$ , we will use the procedure **Find-Dense-Subgraph**( $G_i$ ) (described later). The worst case running time involves  $p(n)$  calls to an algorithm for computing the densest subgraph, which in turn requires  $O(\log n)$  calls to a min-cut/max-flow algorithm (worst case  $O(n^3)$ ). This gives rise to a polynomial time algorithm, albeit with a rather high polynomial complexity. Luckily, in practice, the subgraphs we run the computation on are significantly smaller than the entire graph, so the algorithm runs fairly quickly.

We now show how to generate a polynomial collection of subgraphs  $G_i$ 's satisfying the two properties: distance and subset property. Let  $z$  be a small constant. Consider every subset  $Y_p$  of  $A$  such that  $|Y_p| = z$ , and for all  $t'_A \leq \tau_A$  let  $A_p$  be defined as  $\{v \in A | \forall r \in Y_p \ d_A(v, r) \leq t'_A\}$ . Similarly we define a collection of subsets  $B_q$ : Consider every subset  $Z_q$  of  $B$  such that  $|Z_q| = z$ , and  $t'_B \leq \tau_B$ , let  $B_q$  be defined as  $\{v \in B | \forall r \in Z_q \ d_B(v, r) \leq t'_B\}$ . We generate subgraphs defined by every  $A_p$  and  $B_q$  pair.

First note that, since we are considering every subset  $Y_p \subseteq A$  and  $Z_q \subseteq B$ : the *subset* property holds; namely that one of these subgraphs is guaranteed to contain the optimal subset of nodes. The main difficulty is to show the distance property, that is to show the pairwise distance between every pair of nodes in  $A_p$  is at most  $t'_A$  and the pairwise distance between every pair of nodes in  $B_q$  is at most  $t'_B$ . Now, we exhibit some classes of graphs for which the distance property holds.

**Tree Metric.** Let  $T_A, T_B$  be the trees for  $A$  and  $B$  respectively in  $H$ . The distance in  $T_A(T_B)$  induces the metric  $d_A(d_B)$ .

In this case we need  $z = 2$  (recall that  $z$  is the cardinality of  $Y_p$  and  $Z_q$ ). Choose two vertices  $a$  and  $b$  from  $T_A$  of distance, say  $t'_A$ . and two vertices  $c, d$  from  $T_B$  of distance  $t'_B$ . Define  $Y_p = \{a, b\}$  and  $Z_q = \{c, d\}$ . Obtain the sets  $A_p$  and  $B_q$  as described in the previous subsection. Construct a subgraph induced on the vertex sets  $A_p, B_q$  and obtain the densest subgraph. Return the densest subgraph obtained from all of these subgraphs by making all possible choices for  $\{a, b\}$  and  $\{c, d\}$ .

Now, we prove that the above algorithm (call it **Tree-Densest-Subgraph**) produces an optimum solution, by showing that the distance property holds.

**Theorem 2. Tree-Densest-Subgraph** gives an optimum solution, when  $d_A$  and  $d_B$  form a tree metric.

*Proof.* We only need to show the distance property, that is all the vertices chosen in  $A_p$  have pair-wise distance  $\leq t'_A$  and all the vertices chosen in  $B_q$  have pair-wise distance  $\leq t'_B$ . Pick any two arbitrary vertices  $x, y \in A_p$ . Therefore they are both at distance at most  $t'_A$  from  $a$  and  $b$ . Let the path from  $x$  to  $a$ ,  $P_{x,a}$  intersect  $P_{a,b}$  at  $c_1$  and similarly the path from  $y$  to  $a$  intersect  $P_{a,b}$  at  $c_2$ . Let  $d(x, c_1) = d_1$  and  $d(y, c_2) = d_2$ . Without loss of generality, assume,  $c_1$  is closer to  $a$  than  $c_2$ . Let  $d(a, c_1) = r_1$ ,  $d(c_1, c_2) = r_2$ ,  $d(c_2, b) = r_3$ . Hence the distance between  $x$  and  $y$  is  $d(x, y) = d_1 + r_2 + d_2$ . We have the following sets of equations,  $d_1 \leq r_1$ , otherwise  $x, b$  is a furthest pair. Similarly,  $d_2 \leq r_3$ , otherwise  $(a, y)$  is a furthest pair. Hence  $d_1 + r_2 + d_2 \leq r_1 + r_2 + r_3 \leq t'_A$ . Therefore, all the vertices chosen from  $A_p$  satisfies the distance threshold. Same argument works for vertices chosen from  $B_q$ . Thus the distance property is established.  $\square$

**Some Other Distance Metrics.** The same approach can be extended for graphs where each edge can participate in at most one or two cycles and the problem can be solved optimally in polynomial time. In general it may be possible to extend this approach to graphs where each edge participates in constant number of cycles. The proof technique is similar to the case of trees. Another class of graphs for which we can obtain polynomial time algorithm is *interval* graphs. The proofs of these results can be found in an extended version [29].

### 2.2 Generalization to Arbitrary Graphs

For general graphs, it is not possible to obtain an exact polynomial time algorithm. Here we describe two methods that we implemented. The first method guesses a vertex  $a \in G_S$  from GO ( $b \in G_S$  from PO) and selects all the vertices within distance say  $\frac{t'_A}{2}(\frac{t'_B}{2})$  of  $a(b)$ . Suppose that the set of vertices are denoted by  $X_a(X_b)$ . We now run the algorithm **Find-Dense-Subgraph**( $X_a \cup X_b$ ). This ensures that the vertices are all close to each other, but we may not find the densest subgraph due to the shorter distance requirement.

The second method is identical except that we guess a node  $a$  from GO and a node  $b$  from PO and select all the vertices within distance say  $t'_A(t'_B)$  of  $a(b)$ . Now clearly,  $V(G_S) \subseteq X_a \cup X_b$  and any two vertices in  $X_a$  have distance at most  $2t'_A$  and any two vertices in  $X_b$  have distance at most  $2t'_B$ . Thus if the optimum solution has density  $d_S$  with distance threshold  $t$ , then we guarantee obtaining a subgraph with density at least  $d_S$  and distance at most  $2t$ .

## 3 Densest Subgraphs with a Specified Subset

In this section, we describe the densest subgraph algorithm, where a subset of GO and PO nodes are given apriori and must appear in the returned solution. A distance threshold may also be specified. In that case, we force the subset

of nodes that must appear in the solution, into  $G_i$  and obtain the rest of the vertices by proper guessing as has been shown in the previous section. Thus in this section, we just consider the problem of finding a densest subgraph of a graph when a subset of nodes must appear in the solution.

Given a graph  $G = (V, E)$  and a weight function on the edges  $w'$  and a weight function on the vertices  $w$ , and a subset  $C$  of vertices, we wish to compute a densest subgraph that contains  $C$ . The density of a subgraph is defined as the ratio of the total weight of the edges in the induced graph, to the weights of the nodes in the subgraph (in the unweighted case, all weights are 1). If  $S$  is a subset of nodes then  $E(S)$  is the subset of edges in the subgraph induced by  $S$ . Let  $w'(E(S)) = \sum_{e \in E(S)} w'(e)$ . For a node  $c$ , let  $w'(S, c) = \sum_{(x,c) \in E(S)} w'(x, c)$ . Let  $\bar{E}(S)$  be the set of edges incident to nodes in  $S$  for any  $S \subset V$ .

We first contract all the nodes in  $C$  to a single node  $c$ . We define  $w(c) = \sum_{i \in C} w(i)$ . All the edges between nodes in  $C$  become a self loop on  $c$  with  $w'(c) = \sum_{(i,j) \in E(C)} w'(i, j)$ .

In other words, we wish to compute a subset  $S \subset V \setminus \{c\}$  such that we maximize the following ratio:  $\frac{w'(E(S)) + w'(c) + w'(S, c)}{w(S) + w(c)}$ .

### 3.1 Algorithm for Densest Subgraph without a Specified Subset

We first discuss the basic algorithm for finding a densest subgraph by a series of max-flow (min-cut) computations [17]. This is the procedure **Find-Dense-Subgraph** mentioned earlier. We guess  $\alpha$ , the density of the maximum density subgraph and then refine our guess by doing a network flow computation. Suppose a subset  $S^*$  exists with density  $\alpha^*$  and this is the maximum density subgraph. Suppose our guess is  $\alpha$ . By appropriately defining a flow network and by examining its min-cut structure we are able to determine if  $\alpha = \alpha^*$ , or  $\alpha < \alpha^*$  or  $\alpha > \alpha^*$ . It is very easy to start the binary search since we have upper and lower bounds on the optimal density  $\alpha^*$  and since all densities are rational numbers, once the interval size drops to below  $\frac{1}{|V|^2}$  we can stop.

We next describe the flow network that is constructed. Create a flow network  $G'$  with a source  $s$  and sink  $t$ . We have a node corresponding to each edge in  $G$  (call this set  $E'$ ) and a node corresponding to each node in  $G$  (call this set  $V'$ ). Add edges from  $s$  to  $e \in E'$  of capacity  $w'(e)$  and an edge from  $v \in V'$  to  $t$  with capacity  $\alpha w(v)$ . Add edges from  $e = (x, y) \in E'$  to both  $x \in V'$  and  $y \in V'$  with capacity  $\alpha$  [1].

If  $C = \emptyset$  then the construction proceeds as follows (original problem). First note that there is a  $s$ - $t$  min-cut of value  $w'(E)$ . Suppose the max density subset has density  $\alpha^*$ . Suppose our guess  $\alpha < \alpha^* = \frac{w'(S^*)}{w(S^*)}$ , then it follows that  $\alpha w(S^*) < w'(S^*)$ .

Now consider an  $s$ - $t$  cut  $(s \cup V_1, t \cup V_2)$  in the flow network  $G'$ , then let  $S = V_1 \cap V'$ . The cut includes all the edges from nodes in  $S$  to  $t$  of capacity  $\alpha w(S)$

---

<sup>1</sup> If  $E'$  is a set of hyper-edges then we add such edges from  $e$  to all  $x \in V'$  such that  $x \in e$ .

as well as edges from  $s$  to nodes in  $E'$  that are not in  $V_1$ . All edges  $e = (x, y) \in E$  with one end in  $V \setminus S$  must be in  $V_2$  since otherwise there will be an edge of  $\infty$  capacity across the cut. All the edges in the induced graph formed by  $S$  must be in  $V_1$ , since otherwise we can reduce the capacity of the cut. The weight of this cut is exactly  $w'(E \setminus E(S)) + \alpha w(S)$ . Note that  $w'(E \setminus E(S))$  includes the weight of all edges that are incident on some node in  $E \setminus S$ . The weight of this cut is exactly  $w'(E \setminus E(S)) + \alpha w(S) = w'(E) - w'(S) + \alpha w(S) = w'(E) - (w'(S) - \alpha w(S))$ . But for the optimal subset  $S^*$ , we have  $w'(S^*) - \alpha w(S^*) > 0$  thus there is a cut of value  $< w'(E)$ . So if this happens we know that our guess for  $\alpha$  is  $< \alpha^*$ . Similarly, when our guess for  $\alpha$  is  $> \alpha^*$  then the (unique) min-cut has value  $w'(E)$ . When we make the correct guess, then there are multiple min-cuts of value  $w'(E)$ . Any min-cut other than the trivial gives the correct solution.

### 3.2 Algorithm for Densest Subgraph with a Specified Subset $C$

We now show how to modify this construction when  $C \neq \emptyset$ . We create a new source  $s'$  and add an edge to  $s$  with capacity  $w'(E) - \alpha w(c)$ . We also remove  $c$  from  $V'$ . Again suppose that  $\alpha < \alpha^*$ . In this case, a subset  $S^*$  exists such that  $\frac{w'(E(S^*)) + w'(c) + w'(S^*, c)}{w(S^*) + w(c)} > \alpha$ . Thus,  $w'(E(S^*)) + w'(c) + w'(S^*, c) - \alpha(w(S^*) + w(c)) > 0$ . Rreplace  $w'(E) - w'(\overline{E}(V \setminus (S^* \cup c)))$  for the first term. (Note that  $w'(\overline{E}(V \setminus (S^* \cup c)))$  includes all edges incident on nodes in  $V \setminus (S^* \cup c)$ , and not only the edges induced by those nodes). We now obtain:

$$\begin{aligned}
 &w'(E) - w'(\overline{E}(V \setminus (S^* \cup c))) - \alpha(w(S^*) + w(c)) > 0. \\
 &(w'(E) - \alpha w(c)) - (w'(\overline{E}(V \setminus (S^* \cup c))) + \alpha w(S^*)) > 0. \\
 &(w'(E) - \alpha w(c)) > w'(\overline{E}(V \setminus (S^* \cup c))) + \alpha w(S^*).
 \end{aligned}$$

This means that a min-cut exists (defined by the subset  $S^*$  for example) that is smaller than  $w'(E) - \alpha w(c)$ .

So again by looking at the min-cut structure we should be able to know that  $\alpha < \alpha^*$ . If  $\alpha > \alpha^*$  then the trivial min-cut separating  $s'$  from the rest of the graph is unique. A binary search for  $\alpha$  can be done.

**Side Note:** A simple method that will *not* work is to snap the edges to  $c$  as self loops and to then compute the densest subgraph in  $G$  with  $c$  removed. If the density of the densest subgraph found is lower than  $w'(c)/w(c)$  then we just return  $C$  as the answer. Otherwise we return  $S \cup C$ . The main problem is that the density of  $S$  could get lowered when we merge with  $C$ . The level of dilution depends on the size of the densest subgraph in  $G$  with  $c$  removed; hence a subgraph with slightly lower density than the optimal solution, but of much larger size could be a better choice.

## 4 Finding All almost Maximum Densest Subgraphs

In this section, we describe an algorithm for computing all densest subgraphs as well as all almost maximum densest subgraphs. It might not be sufficient just

to find only one subgraph of highest density, and sometimes subgraphs having density close to the maximum might be interesting as well. If  $\alpha^*$  is the highest density, our goal is to find all subgraphs that have density close to  $\alpha^*$ . A subgraph  $S$  that has density  $\alpha^*(1 - \delta_S)$  is lacking by a factor of  $(1 - \delta_S)$  from the optimum. Thus if we want to detect it, we have to relax the density requirement of  $S$  by a  $(1 - \delta_S)$  factor. The amount of relaxation may differ depending on the size of the returned subgraph. We denote by  $D(S)$  the density of the subgraph induced by  $S$ .

Formally, given a graph  $G = (V, E)$ , if  $\alpha^* = \max_{S \subseteq V} D(S)$ , then given an  $\epsilon > 0$ , we want to return  $T = \{S \mid S \subseteq V, D(S) \geq \left(1 - \frac{\epsilon}{|S|}\right) \alpha^*\}$ . Therefore, we have  $\delta_S = \frac{\epsilon}{|S|}$ . We consider the unweighted case, where vertices and edges all have unit weights. Extension to arbitrary weight is trivial. Also we can pose the distance restriction as in Section 2 easily and get the same approximation results as we obtained earlier.

Recall the construction of flow network from Section 3. We guess  $\alpha$  as the value of density and create a flow network  $N(G)$  for graph  $G$ . If  $\{s \cup V_1, t \cup V_2\}$  is the minimum cut and  $V_1 \wedge V = S$ , then the value of min-cut is  $K = |E| - (E(S) - \alpha|S|)$ . Thus when,  $\alpha = \alpha^*$ ,  $K = |E|$ . The algorithm searches for the value of  $\alpha^*$  using a binary search. Since the gap between two consecutive density values, is at least  $\frac{1}{|V|^2}$  [12], the value of  $\alpha^*$  can be guessed accurately in  $O(\log n)$  time.

We construct the flow network  $N_{\alpha^*}$  with  $\alpha^*$  as the guess, and compute all min-cuts having value  $\leq |E| + \epsilon\alpha^*$ . There are two questions, “how can we compute all min-cuts of value  $\leq |E| + \epsilon\alpha^*$  ?” and “how can  $T$  be detected from these min-cut computations ?”. While we address the first question in Subsection 4.1, following lemma answers the second.

**Lemma 1.** *Let  $M = \{V_1 \mid \text{cut}(s \cup V_1, t \cup (V \setminus V_1)) \leq |E| + \alpha^*\epsilon\}$ , then  $T = \{V_1 \cap V\}$ .*

*Proof.* Let  $S' \in T$  and  $S' = V_1' \wedge V$ . Then the cut induced by  $s \cup V_1'$  is  $|E| - (E(S') - \alpha^*|S'|) = |E| - |S'|(D(S') - \alpha^*)$ . Since  $S' \in T$ ,  $D(S') \geq \alpha^*(1 - \frac{\epsilon}{|S'|})$ . Thus, the cut induced by  $s \cup V_1'$  is at most  $|E| - |S'|(\alpha^*(1 - \frac{\epsilon}{|S'|}) - \alpha^*) = |E| + \alpha^*\epsilon$ . Hence,  $V_1' \in M$ . On the other hand, if  $V_1' \in M$ , then the cut value of  $s \cup V_1'$  is,  $|E| - (E(S') - \alpha^*|S'|) \leq |E| + \alpha^*\epsilon$ . Thus,  $\alpha^*|S'| - E(S') \leq \alpha^*\epsilon$ , or  $D(S') \geq \alpha^*(1 - \frac{\epsilon}{|S'|})$ . □

Now we show how by modifying Picard and Queyranne’s algorithm [25], we can compute all cuts of value  $\leq |E| + \epsilon\alpha^*$  in  $N_{\alpha^*}$ .

### 4.1 Finding All Almost Min-Cuts

In Picard’s algorithm, we are given a finite directed network  $N = (V, E, c)$ , with vertex set  $V$ , including a source  $s$  and a sink  $t$ , arc sets  $E$  and positive capacities  $c_{i,j}$  defined on every  $(i, j) \in E$ . The goal is to compute all  $s$ - $t$  cuts having minimum value. Given a binary relation  $R$  on  $V$ , a subset  $C \subseteq V$  is said

to be closure for  $R$ , iff for all vertices  $i, j \in V$ , the conditions  $i \in C$  and  $iRj$  imply  $j \in C$ . Picard showed that, if  $f$  is a maximum flow in  $N$ ,  $cres$  is the residual capacity and  $R$  is defined as,  $iRj$ , iff  $cres(i, j) > 0$ , then a cut  $(S, \bar{S})$  separating  $s$  from  $t$  is a minimum cut iff  $S$  is a closure for  $R$  containing  $s$  and not  $t$ . By enumerating all closures of  $R$ , all the min-cuts can now be detected.

We define the relation  $R$  as,  $iRj$  iff  $cres(i, j) > \epsilon\alpha^* = \delta$  instead of  $cres(i, j) > 0$ . The following lemma connects all almost  $s$ - $t$  cuts with the closures for  $R$ .

**Theorem 3.** *All  $s$ - $t$  cuts of value  $\leq K + \delta$  are closures for  $R$ , where  $K$  is the value of minimum  $s$ - $t$  cut.*

*Proof.* Consider a cut  $S$  of value  $K' \leq K + \delta$ . Let the edges across the cut  $E(S, \bar{S}) = \{e_1, e_2, \dots, e_l\}$ ,  $e_i \in E(G)$  with capacities  $\{c_1, c_2, \dots, c_l\}$  and flow  $\{f_1, f_2, \dots, f_l\}$ . Since maxflow is equal to min-cut, when we consider the max flow in the network, we must have,  $f(S, \bar{S}) - f(\bar{S}, S) = K$ . Hence  $f(S, \bar{S}) > K$ . Let if possible one of the residual capacity, say of  $e_1$  be higher than  $\delta$ , then that will imply  $S$  is not a closure for  $R$ . We have  $K < f_1 + f_2 + \dots + f_l < (c_1 - \delta) + c_2 + \dots + c_l = K' - \delta$ . So  $K' > K + \delta$ , giving a contradiction.  $\square$

Therefore, we again enumerate all the closures, and discard any closure for which cut value is  $> K + \delta$ . This last step is necessary, since there can be some closures for  $R$  that do not necessarily give a cut of value  $\leq K + \delta$ . The closures for  $R$  contain all the cuts of value  $\leq K + \delta$  and some cuts of value  $K + \delta(l + l')$ .

## 5 Experiments on the TAIR Dataset

We briefly summarize the results of several experiments. In a first experiment (dataset  $SD_1$ ) we analyze 10 photomorphogenesis genes. We use the literature to validate patterns identified in a dense subgraph. We highlight some interesting patterns that are novel and could lead to new hypotheses. A control experiment (dataset  $SD_2$ ) includes the 10 photomorphogenesis genes and 10 additional control genes. The control experiment is used to confirm that all patterns identified in the dense subgraphs are true positives and are validated in the literature. There were no false positive patterns identified in our experiment. In a subsequent experiment (dataset  $SD_3$ ), we analyze 20 genes involved in different (and currently unrelated) biological pathways. We also perform experiments to study the properties of the dense subgraphs, for different experiment protocols and parameters such as the number of genes and the GO and PO distance thresholds. These results are in an extended report [29].

We execute a protocol to retrieve all TAIR genes, their GO and PO annotations, and the reference publications from PubMed. As of January 2009, the LSLink TAIR dataset contains 3540 GO CV terms, 350 PO CV terms, 18861 genes, 70128 GO annotations, 484261 PO annotations and 1873250 (GO, PO) pairs. The average number of GO annotations and PO annotations for the TAIR genes is 3.97 and 3.13, respectively. The maximum number of annotations for any gene is 22 GO annotations and 50 PO annotations.

## 5.1 Photomorphogenesis Case Study

We report on promising results of a photomorphogenesis case study on dataset  $SD_1$  with the following 10 TAIR genes: CRY1, CRY2, HFR1, CIB1, CIB5, SHB1, COP1, HY5, PHOT1, PHOT2. These 10 genes are associated with 107 annotations (66 GO terms and 41 PO terms) and 2230 combinations of (GO, PO) terms. The edge weight from the corresponding bipartite graph ranged from a minimum of 1 (1368 edges) to a maximum of 7 (2 edges). We applied the *distance restricted dense subgraph* algorithm with GO distance threshold of 2 and PO distance threshold of 3, which identified a complex pattern involving the following subset: 9 genes, CRY1, CRY2, HFR1, CIB5, COP1, HY5, PHOT1, PHOT2, SHB1 3 GO terms, 5634: nucleus; cellular\_component, 5773: vacuole; cellular\_component, and 5794: Golgi apparatus; cellular\_component; and 13 PO terms. These obtained GO and PO terms are shown in Figure 3. Figure 3 also shows a subgraph chosen from the densest subgraph involving 2 GO terms, 5634: nucleus; cellular\_component and 5773: vacuole; cellular\_component; and 2 PO terms, 13: cauline leaf; plant\_structure and 37: shoot apex; plant\_structure are shown in Figure 3. This creates 4 (GO, PO) pairs as follows: (5634, 13); (5634, 37); (5773, 13); (5773, 37). Figure 3 also illustrates the genes that are annotated by these pairs.

We make the following observations:

- The combinations of (GO,PO) edges observed in this complex pattern are consistent with the literature and provides validation that the complex pattern is meaningful. Details of all the observations can be found in an extended version [29].
- Specific combinations of genes and (GO,PO) edges are interesting in that they can lead to further hypothesis. We identify 5 potentially interesting patterns from the subgraph. We elaborate on two patterns. Details can be found in an extended version [29].
- HFR1 is not annotated with the following GO and PO combination: (5634: nucleus; cellular\_component and 37: shoot apex; plant\_structure). This is indicated by an arrow in Figure 3. A review of the literature suggests that this is a novel observation about the mechanism controlling this gene that should be pursued further.
- The next observation confirms the potential benefits of our approach to finding complex patterns in the annotated *LSLink* datasets. Consider the pattern of annotation that includes the 2 genes CRY2 and PHOT1. Both are annotated with the following 2 GO and PO combinations: (5773: vacuole; cellular\_component and 13: cauline leaf; plant\_structure) and (5773: vacuole; cellular\_component and 37: shoot apex; plant\_structure). These annotations are also marked with an arrow in Figure 3. We observe that there are only 2 papers in the literature, [23] published in 2004, and [13] published in 2008, that postulate that some members of the CRY and PHOT families may be functionally interactive in vacuoles. Indeed, these two papers came to this conclusion only after significant experimental research.



Gene	(GO PO) edge			
	5634-13	5634-37	5773-1	5773-37
HFR1 (AT1G02340)	1	0	0	0
CRY2 (AT1G04400)	1	1	1	1
CIB5 (AT1G26260)	1	1	0	0
COP1 (AT2G32950)	1	1	0	0
PHOT1 (AT3G45780)	0	0	1	1
CRY1 (AT4G08920)	1	1	0	0
SHB1 (AT4G25350)	1	0	0	0
HY5 (AT5G11260)	1	1	0	0
PHOT2 (AT5G58140)	0	0	0	0
CIB1 (AT4G34530)	0	0	0	0

Fig. 3. Potential Complex Pattern of Photomorphogenesis Genes

To summarize, our user evaluation confirmed the benefit of using the dense subgraph approach of identifying complex patterns based on the underlying patterns of annotation, without having to completely digest the scientific literature and/or complete an experiment protocol.

We performed a control experiment using  $SD_2$ ;  $SD_2$  included the 10 genes of  $SD_1$  and 10 additional genes that were chosen randomly from genes that had some common annotations with the genes in  $SD_1$ . The goal of this experiment is to verify that the pattern emerged from experimenting on  $SD_1$  alone still persists and thus to confirm that it was not a random pattern. The dense subgraph for  $SD_1$  included 9 genes, 3 GO terms, 13 PO terms and 39 (GO, PO) edges. The dense subgraph for  $SD_2$  included 14 genes, 4 GO terms, 11 PO terms and 44 (GO, PO) edges. The genes included the 8 photomorphogenesis genes (HFR1 CRY2 CIB5 COP1 PHOT1 CRY1 SHB1 HY5) and 6 control genes (GAPC2 FT ARF3 AG ARF4 REV). The gene PHOT2 is not included. Further, the GO term Golgi apparatus and 3 PO terms cauline leaf, leaf whorl and petiole were not present. Two GO terms mitochondrion and cytosol and 1 PO term inflorescence meristem were introduced. Detailed observations from the control dense subgraph for  $SD_2$  can be found in an extended version [29].

While the control dense subgraph for  $SD_2$  does show some variations in terms of photomorphogenesis genes, GO and PO terms from that obtained using  $SD_1$ , we verified that none of these variations are significant, i.e., the variations do not contradict any of the patterns of annotation of the dense subgraph for  $SD_1$ . Further the patterns of  $SD_1$ , that were found validating the literature or can lead to potentially new hypothesis are unchanged. For example, PHOT2 which is excluded from the control subgraph, as well as the GO and PO terms that are excluded were not included in any of the  $SD_1$  patterns. An unexpected benefit is that the control densest subgraph for  $SD_2$  was itself able to yield some interesting patterns that could lead to new hypotheses.

We note that developing a NULL hypothesis to test the significance of the dense subgraphs that we generate is non trivial since there are many metrics to compare the similarity of two graphs. One option is to add control genes as described. Other alternatives include comparing the density distribution of dense subgraphs from a random graph versus the dense subgraphs from our datasets.



Another would generate *all almost dense subgraphs* to determine if they are different with respect to both metrics as well as the observed patterns. One may also consider random labeling of the GO and PO terms in the datasets. We will explore these alternatives in future work.

Additional experiments on different set of genes and empirical study on the properties of the densest subgraph algorithms can be found in an extended version [29].

**Acknowledgments.** We thank Carl Kingsford and Mihai Pop for useful discussions about our results and their feedback has been invaluable.

## References

1. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4 (2003)
2. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research* 32(Database issue), 267–270 (2004)
3. Charikar, M.: Greedy approximation algorithms for finding dense components in a graph. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 84–95. Springer, Heidelberg (2000)
4. Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families 30(7), 1575–1584 (April 2002)
5. Entrez: the life sciences search engine, <http://www.ncbi.nih.gov/gquery/gquery.fcgi>
6. Sayers, E.W., et al.: Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research* 37(Database issue), D16–D18 (2009)
7. Ashburner, M., et al.: Gene Ontology: tool for the unification of biology. *Nature Genetics* 25(1), 25–29 (2000)
8. Margarita, et al.: TAIR: a resource for integrated Arabidopsis data. *Functional and Integrative Genomics* 2(6), 239 (2002)
9. Rhee, S.Y., et al.: The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to arabidopsis biology, research materials and community. *Nucleic Acids Research* 31(1), 224–228 (2003)
10. Feige, U.: A threshold of  $\ln n$  for approximating set cover. *Journal of the ACM* 45(4), 634–652 (1998)
11. Gene Ontology (GO), <http://www.geneontology.org/>
12. Goldberg, A.V.: Finding a maximum density subgraph. Technical report (1984)
13. Kang, B., Grancher, N., Koyffmann, V., Lardemer, D., Burney, S., Ahmad, M.: Multiple interactions between cryptochrome and phototropin blue-light signalling pathways in arabidopsis thaliana. *Planta* 227(5), 1091–1099 (2008)
14. Khuller, S., Saha, B.: On finding dense subgraphs. In: ICALP 2009, pp. 597–608 (2009)
15. King, A.D., Przulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. *Bioinformatics* 20(17), 3013–3020 (2004)
16. Rhee, S.Y., Reiser, L.: Using The Arabidopsis Information Resource (TAIR) to Find Information About Arabidopsis Genes. *Current Protocols in Bioinformatics* (2005)

17. Lawler, E.: Combinatorial optimization - networks and matroids. Holt, Rinehart and Winston, New York (1976)
18. Lee, W.-j., Raschid, L., Sayyadi, H., Srinivasan, P.: Exploiting ontology structure and patterns of annotation to mine significant associations between pairs of controlled vocabulary terms. In: Bairoch, A., Cohen-Boulakia, S., Froidevaux, C. (eds.) DILS 2008. LNCS (LNBI), vol. 5109, pp. 44–60. Springer, Heidelberg (2008)
19. Li, X., Foo, C., Ng, S.: Discovering protein complexes in dense reliable neighborhoods of protein interaction networks 6, 157–168 (2007)
20. Maglott, D.R., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research* 35(Database issue), 26–31 (2007)
21. Navlakha, S., White, J., Nagarajan, N., Pop, M., Kingsford, C.: Finding biologically accurate clusterings in hierarchical tree decompositions using the variation of information. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 400–417. Springer, Heidelberg (2009)
22. Newman, M.E.J.: Modularity and community structure in networks 103(23), 8577–8582 (2006)
23. Ohgishi, M., Saji, K., Okada, K., Sakai, T.: Functional analysis of each blue light receptor, cry1, cry2, phot1, and phot2, by using combinatorial multiple mutants in arabidopsis. *PNAS* 101(8), 2223–2228 (2004)
24. Pereira-Leal, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. *Proteins* 54(1), 49–57 (2004)
25. Picard, J.-C., Queyranne, M.: On the structure of all minimum cuts in a network and applications. *Mathematical Programming Study* 13, 8–16 (1980)
26. Plant Ontology (PO), <http://www.plantontology.org/>
27. PubMed, <http://www.ncbi.nih.gov/entrez/>
28. PubMed Central, <http://www.pubmedcentral.nih.gov/>
29. Saha, B., Hoch, A., Khuller, S., Raschid, L., Zhang, X.: Dense subgraph with restrictions and applications to gene annotation graphs (2010), <http://www.cs.umd.edu/~samir/grant/recomb-full.pdf>
30. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks 100(21), 12123–12128 (October 2003)
31. Unified Medical Language System (UMLS), <http://www.nlm.nih.gov/research/umls/>
32. Yu, H., Paccanaro, A., Trifonov, V., Gerstein, M.: Predicting interactions in protein networks by completing defective cliques. *Bioinformatics* 22(7), 823–829 (2006)

# Time and Space Efficient RNA-RNA Interaction Prediction via Sparse Folding

Raheleh Salari<sup>1,\*</sup>, Mathias Möhl<sup>2,\*</sup>, Sebastian Will<sup>2,\*</sup>,  
S. Cenk Sahinalp<sup>1,\*\*</sup>, and Rolf Backofen<sup>2,\*\*</sup>

<sup>1</sup> Lab for Computational Biology, School of Computing Science,  
Simon Fraser University, Burnaby, BC, Canada  
{rahelehs, cenk}@cs.sfu.ca

<sup>2</sup> Bioinformatics, Institute of Computer Science, Albert-Ludwigs-Universität,  
Freiburg, Germany  
{mmohl, will, backofen}@informatik.uni-freiburg.de

**Abstract.** In the past years, a large set of new regulatory ncRNAs have been identified, but the number of experimentally verified targets is considerably low. Thus, computational target prediction methods are on high demand. Whereas all previous approaches for predicting a general joint structure have a complexity of  $O(n^6)$  running time and  $O(n^4)$  space, a more time and space efficient interaction prediction that is able to handle complex joint structures is necessary for genome-wide target prediction problems. In this paper we show how to reduce both the time and space complexity of the RNA-RNA interaction prediction problem as described by Alkan et al. [1] via dynamic programming sparsification - which allows to discard large portions of DP tables without losing optimality. Applying sparsification techniques reduces the complexity of the original algorithm from  $O(n^6)$  time and  $O(n^4)$  space to  $O(n^4\psi(n))$  time and  $O(n^2\psi(n) + n^3)$  space for some function  $\psi(n)$ , which turns out to have small values for the range of  $n$  that we encounter in practice. Under the assumption that the polymer-zeta property holds for RNA-structures, we demonstrate that  $\psi(n) = O(n)$  on average, resulting in a linear time and space complexity improvement over the original algorithm. We evaluate our sparsified algorithm for RNA-RNA interaction prediction by total free energy minimization, based on the energy model of Chitsaz et al. [2], on a set of known interactions. Our results confirm the significant reduction of time and space requirements in practice.

## 1 Introduction

Starting with the discovery of microRNAs (miRNAs) and the advent of genome-wide transcriptomics, it has become clear that RNA plays a large variety of important roles in living organisms that extend far beyond being a mere intermediate in protein biosynthesis [3]. Several of these non-coding RNAs (ncRNAs) regulate gene expression post-transcriptionally through base pairing (and establishing a joint structure) with a target mRNA, as per the eukaryotic miRNAs

---

\* Joint first authors.

\*\* To whom correspondence should be addressed.

and small interfering RNAs (siRNAs) [4,5,6], antisense RNAs [7,8] or bacterial small regulatory RNAs (sRNAs) [9]. In addition to such endogenous regulatory ncRNAs, antisense oligonucleotides have been used as exogenous inhibitors of gene expression; antisense technology is now commonly used as a research tool as well as for therapeutic purposes. Furthermore, synthetic nucleic acids systems have been engineered to self assemble into complex structures performing various dynamic mechanical motions [10,11,12,13,14].

Despite all the above advances, the first set of computational methods for predicting ncRNA-target mRNA interactions suffered from over-simplifying the types of interactions allowed. As a result they could not accurately predict many known interactions, especially those involving long ncRNAs. More precisely, these methods either restricted the interactions to external positions, or they allowed interactions with at most one interaction site. These restrictions were lifted by two independently developed methods, which provided the first set of algorithms for predicting a precise interaction structure of two RNA strands: (i) the algorithm by Pervouchine [15], for example, maximizes the total number of base pairs, and (ii) a more general method by Alkan et al. [1], minimizes the total free energy of the interacting RNA strands using a nearest neighbor energy model. Alkan *et al.* also provide a proof of the NP-completeness of the general problem, together with a precise definition of interaction types that can be handled, as well as the first experimental confirmation of the total free energy minimization approach via correctly predicting the joint structure formed by a number of interacting RNA pairs.

More recently, two approaches [2,16] independently solved the problem of calculating the partition function for the interaction model introduced by Alkan *et al.*, allowing to determine important thermodynamic quantities like melting temperatures. As demonstrated in [2], the computed melting temperatures are in a good agreement with experimentally measured ones.

One key problem with the above approaches for predicting a general joint structure [15,1,2,16] is that they all have a worst case running time of  $O(n^6)$  and a space complexity of  $O(n^4)$ . While this complexity might be acceptable when analyzing only a few putative sRNA-target interaction pairs, we are now faced with the situation that the amount of data to be analyzed is vastly increasing. To give an example, a recent mapping of transcripts using tiling arrays in the budding yeast *S. cerevisiae* [17] with 5,654 annotated open reading frames (ORF) has found 1555 antisense RNAs that overlap at least partially with the ORFs at the opposite strand. Currently, it is completely unclear what these antisense RNAs are doing - whether they target only their associated sense mRNA or have also other mRNA targets, and whether they always form a complete duplex or more complex joint structures such as multiple kissing hairpins if they overlap only partially is not known. The same situation appears in many other species. Thus, there is urgent need for a more time and space efficient interaction prediction method that is able to handle complex joint structures.

In this paper we present a new method for calculating the joint structure of interacting RNAs by minimizing their total free energy, which improves time

and space efficiency over previous approaches. As first in its class, the method is sufficiently fast to be applied in large scale screening approaches. We suggest to refine putative interacting pairs with even more accurate RNA-RNA-interaction prediction approaches [2116]. Because these approaches compute a partition function for RNA-RNA-interaction, they can determine important thermodynamic parameters such as melting temperatures, however their efficiency cannot be improved in the same way.

We show how to reduce both time and space complexity using an approach called *sparsification*, which uses the observation that the resulting DP-matrices are sparse. As previous applications of sparsification to problems related to RNA folding, our approach exploits a triangle inequation on the dynamic programming matrix. Assuming the *polymer-zeta* property for interacting RNAs, we show an efficiency gain by a linear factor. This *polymer-zeta* property basically states that the probability of a base pair decreases with its size, i.e. there are only few long range base pairs.

In this paper we consider a version of the polymer-zeta property for interacting RNAs and develop novel algorithmic approaches as (1) we cannot assume the standard polymer-zeta property for all base pairs as for intermolecular base pairs there is no clear notion of a distance between the bases; (2) the joint interaction prediction problem does not allow to split only at arcs in the recursion, which was crucial in the demonstration of a linear (asymptotic) speed up for problems involving the folding of a single RNA.

We sparsify the dynamic programming tables involved in total free energy minimization first described in Alkan *et al.* [1] on the more general energy model of Chitsaz *et al.* [2] resulting in a significant reduction in time and space complexity. There are four different cases that need to be sped up, which results in a total of four different candidate lists; for each sequence and each region, we have to consider folding with interaction or without interaction, which gives rise to two candidate lists per sequence. We emphasize that beyond reducing time complexity, we obtain a similar space reduction even in the intricate setting of four independent candidate lists.

*Sparsification in RNA folding.* The general technique of DP sparsification has been used in the context of RNA-folding, to reduce the time and space complexity of two central problems in this domain, namely (i) the calculation of the MFE structure of a single RNA sequence folding [18][19], and (ii) the Sankoff approach [20] of simultaneous folding and alignment of two RNAs [21][19]. In both cases, a (roughly) linear reduction in the time complexity was achieved on average.<sup>1</sup> The time/space reduction is based on the assumption that RNA-structures or consensus structures - in the simultaneous alignment and folding of RNAs,

<sup>1</sup> To be more precise, the time complexity of RNA-folding was reduced from  $O(n^3)$  to  $O(nZ)$  and the space complexity was reduced from  $O(n^2)$  to  $O(Z)$ , where  $Z$  is a sparsity factor satisfying  $n \leq Z \leq n^2$ . An estimation [18] of the expected value of a parameter related to  $Z$ , based on a probabilistic model for polymer folding and measured by simulations, shows that  $Z$  is significantly smaller than  $O(n^2)$ . Similar results are given for the co-folding problem.

satisfy the polymer-zeta behavior, which is an assumption that we employ in predicting the intramolecular base-pairs observed in RNA joint structures. The above approaches for RNA folding as well as simultaneous folding and alignment use the polymer-zeta property for either a single RNA sequence and structure, or for a consensus structure of two (structurally similar) RNAs, leading to a single candidate list.

*RNA-RNA interaction prediction methods.* The first set of computational methods to calculate joint structures formed by interacting RNAs (e.g., RNAhybrid [22] or TargetRNA [23]) considered only the base-pairs between the two different strands that form a duplex structure. Since this ignores the intramolecular structures, later approaches aimed to predict a joint structure for both interacting RNAs. This second generation of RNA-RNA interaction prediction methods, which include pairfold [24], RNAcofold [25] and the method presented by Dirks *et al.* as part of the NUpack package [26], consider joint structures of mRNA and sRNA that are generated by concatenating the two sequences using a special linker character. Then, a modified version of the standard RNA-folding algorithms (such as Mfold [27] or RNAfold [28]) which preserve the basic recursive structure of standard RNA-folding but specially treat loops that contain the linker symbol, is applied. Unfortunately, none of the above approaches can predict joint structures with kissing hairpin interactions. For that reason, a third generation of RNA-RNA interaction prediction algorithms (in particular, RNAup [29] and IntaRNA [30]) were recently introduced. These approaches first determine the accessibility of all putative interaction sites, from which an energy to make the sites free of intramolecular base-pairs can be calculated. Later, this energy is combined with the energy of the duplex that can be formed between different interaction sites.

Clearly, the third generation methods can only handle one interaction site per sequence - which may not include any intramolecular base-pairs. As a result, two or more kissing hairpins as per the interaction between OxyS and fhlA [31] cannot be treated by these approaches. For the purpose of handling such complex joint structures, more sophisticated DP-methods of Pervouchine [15] and Alkan *et al.* [1], as well as the partition function variants by Chitsaz *et al.* [2] and Huang *et al.* [16] were introduced. Finally, more recent methods introduced in [32,33] can be seen as heuristic approximations to the full model of [2], or as an extension of the accessibility approaches (RNAup/IntaRNA) to several interaction sites.

## 2 Preliminaries

Throughout this paper, we denote the two nucleic acid strands by  $\mathbf{R}$  and  $\mathbf{S}$ . Strand  $\mathbf{R}$  is indexed from 1 to  $L_R$  in 5' to 3' direction and  $\mathbf{S}$  is indexed from 1 to  $L_S$  in 3' to 5' direction. Note that the two strands interact in opposite directions, e.g.  $\mathbf{R}$  in 5'  $\rightarrow$  3' with  $\mathbf{S}$  in 3'  $\leftarrow$  5' direction. Each nucleotide is paired with at most one nucleotide in the same or the other strand. The subsequence from the  $i^{th}$  nucleotide to the  $j^{th}$  nucleotide in a strand is denoted by  $[i, j]$ . We refer to the  $i^{th}$  nucleotide in  $\mathbf{R}$  and  $\mathbf{S}$  by  $i_R$  and  $i_S$  respectively. An intramolecular base

pair between the nucleotides  $i$  and  $j$  in a strand is called an *arc* and denoted by a bullet  $i \bullet j$ . An intermolecular base pair between the nucleotides  $i_R$  and  $i_S$  is called a *bond* and denoted by a circle  $i_R \circ i_S$ . An arc  $i_R \bullet j_R$  (or respectively  $i_S \bullet j_S$ ) *covers* a bond  $k_R \circ k_S$  if  $i_R < k_R < j_R$  (or  $i_S < k_S < j_S$ ). An arc is called *interaction arc* if it covers a bond. A subsequence  $[i_R, j_R]$  (or  $[i_S, j_S]$ , analogously) contains a *direct bond*,  $k_R \circ k_S$ , if  $i_R \leq k_R \leq j_R$  and no arc within  $[i_R, j_R]$  covers  $k_R \circ k_S$ . Two bonds  $i_R \circ i_S$  and  $j_R \circ j_S$  are called *crossing bonds* if  $i_R < j_R$  and  $i_S > j_S$  or  $i_R > j_R$  and  $i_S < j_S$ . An interaction arc  $i_R \bullet j_R$  in  $R$  *subsumes* a subsequence  $[i_S, j_S]$  in  $S$  if there is at least one bond  $k_R \circ k_S$ , where  $i_R < k_R < j_R$  and  $i_S < k_S < j_S$ , and for all bonds  $k_R \circ k_S$ , if  $i_S \leq k_S \leq j_S$  then  $i_R < k_R < j_R$ . Analogously, interaction arcs in  $S$  can subsume subsequences in  $R$ . Two interaction arcs  $i_R \bullet j_R$  and  $i_S \bullet j_S$  are part of a *zigzag*, if there is a bond  $k_R \circ k_S$ , where  $i_R < k_R < j_R$  and  $i_S < k_S < j_S$ , but neither  $i_R \bullet j_R$  subsumes  $[i_S, j_S]$  nor  $i_S \bullet j_S$  subsumes  $[i_R, j_R]$ .

We represent the recursions of our dynamic programming (DP) algorithm in a graphical notation using the recursion diagrams introduced in [2]. Within the recursion diagrams, a horizontal line indicates the phosphate backbone, a solid curved line indicates an arc, and a dashed curved line encloses a region and denotes its two terminal bases which may be paired or unpaired. Letters within a region specify a recursive quantity. White regions are recursed over and blue regions indicate those portions of the secondary structure that are fixed at the current recursion level and contribute to the energy as defined by the energy model. Green and red regions have the same recursion cases as the corresponding white regions, except that for the green regions multiloop energy and for red regions kissing loop energy is applied, i.e. the corresponding penalties for each unpaired base and base pair should be applied. A solid vertical line indicates a bond, a dashed vertical line denotes two terminal bases of a region which may be base paired or unpaired, and a dotted vertical line denotes two terminal bases of a region which are assumed to be unpaired. A terminal determined by  $\bullet$  is starting point of either an interaction arc or a bond.

### 3 Methods

In this section we discuss an algorithm for RNA-RNA interaction prediction via total free energy minimization, under the assumption that there are no (internal) pseudoknots, crossing bonds (i.e. external pseudoknots), or zigzags in the joint structure. The algorithm is similar to the one introduced by Alkan et al. [1] on a simpler energy model. We use sparsification techniques to reduce the complexity of the original algorithm from  $O(n^6)$  time and  $O(n^4)$  space to  $O(n^4\psi(n))$  time and  $O(n^2\psi(n)+n^3)$  space for some function  $\psi(n) = O(n)$  on average. To simplify the presentation, we discuss the sparsification for the joint structure prediction via total base pair maximization. Note that RNA-RNA interaction based on base pair maximization is the generalized version of the Nussinov model [34] for single RNA folding and was employed by Pervouchine [15] as well as Alkan et al. [1] for RNA-RNA interaction prediction. Later in the paper we also provide



all concepts for generalizing the algorithm to capture a more realistic energy model provided by Chitsaz et al. [2].

### 3.1 Sparsification for Maximizing Base Pairs

Given two RNA sequences  $\mathbf{R}$  and  $\mathbf{S}$ ,  $N(i_R, j_R, i_S, j_S)$  denotes the maximum number of base pairs in the joint structure of  $[i_R, j_R]$  and  $[i_S, j_S]$ , and  $N^{\mathbf{X}}(i, j)$  (for  $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$ ) denotes the maximum number of base pairs of the subsequence  $[i, j]$  of the single sequence  $\mathbf{X}$ . The recursion cases for computing the maximum number of base pairs for RNA-RNA interaction are illustrated in Fig. 1.  $N(i_R, j_R, i_S, j_S)$  and  $N^{\mathbf{X}}(i, j)$  for  $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$  are calculated by the following recursions

$$N(i_R, j_R, i_S, j_S) = \max \left\{ \begin{array}{ll} N(i_R + 1, j_R, i_S, j_S) & (a) \\ N(i_R, j_R, i_S + 1, j_S) & (b) \\ N(i_R + 1, j_R, i_S + 1, j_S) + 1 & (c) \\ \max_{\substack{i_R < k \leq j_R \\ R[i_R], R[k] \text{ compl.}}} \left( \begin{array}{l} 1 + N^{\mathbf{R}}(i_R + 1, k - 1) \\ + N(k + 1, j_R, i_S, j_S) \end{array} \right) & (d) \\ \max_{\substack{i_S < k \leq j_S \\ S[i_S], S[k] \text{ compl.}}} \left( \begin{array}{l} 1 + N^{\mathbf{S}}(i_S + 1, k - 1) \\ + N(i_R, j_R, k + 1, j_S) \end{array} \right) & (e) \\ \max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ R[i_R], R[k_R] \text{ compl.}}} \left( \begin{array}{l} 1 + N(i_R + 1, k_R - 1, i_S, k_S) \\ + N(k_R + 1, j_R, k_S + 1, j_S) \end{array} \right) & (f) \\ \max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ S[i_S], S[k_S] \text{ compl.}}} \left( \begin{array}{l} 1 + N(i_R, k_R, i_S + 1, k_S - 1) \\ + N(k_R + 1, j_R, k_S + 1, j_S) \end{array} \right) & (g) \end{array} \right. \quad (1)$$
  

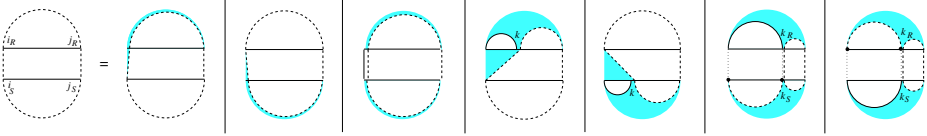
$$N^{\mathbf{X}}(i, j) = \max \left\{ \begin{array}{ll} N^{\mathbf{X}}(i + 1, j) & (a) \\ \max_{\substack{i < k \leq j \\ X[i], X[k] \text{ compl.}}} \left( \begin{array}{l} 1 + N^{\mathbf{X}}(i + 1, k - 1) \\ + N^{\mathbf{X}}(k + 1, j) \end{array} \right) & (b) \end{array} \right. \quad (2)$$

In Eq. 1, the cases (a) and (b) introduce an unpaired base at positions  $i_R$  and  $i_S$  respectively, and case (c) introduces a bond  $i_R \circ i_S$ . Cases (d) and (f) introduce an arc at  $i_R \bullet k$  and cases (e) and (g) at  $i_S \bullet k$ , where cases (f) and (g) assume that the arc is an interaction arc and cases (d) and (e) assume that this is not the case.

**Time reduction by sparsification.** We will apply a sparsification technique to reduce the number of cases necessary to be considered for Eq 1(d)-(g), as well as Eq 2(b).

Concerning sparsification, the simple cases are Eq 1(d),(e), and Eq 2(b), which correspond to the folding of a single sequence. The sparsification of these cases works in close analogy to the sparsification of RNA structure prediction as described by Wexler et al. [18]. We will briefly review their approach adapted to case Eq 2(b). Thereafter, we describe sparsification of the complex cases.





**Fig. 1.** Recursion cases for computing the maximum base pairing joint structure of  $[i_R, j_R]$  and  $[i_S, j_S]$

*Sparsifying recursion cases for single structure folding.* The key to sparsification is a triangle inequality property of the DP matrix. In the case of  $N^{\mathbf{X}}$ , for every subsequence  $[i, j]$  and  $i < k \leq j$  the following inequality holds:

$$N^{\mathbf{X}}(i, j) \geq N^{\mathbf{X}}(i, k) + N^{\mathbf{X}}(k + 1, j).$$

Due to this property, it is sufficient to maximize in Eq. 2(b) for each  $i$  only over certain candidates  $k$  instead of all  $k$  with  $i < k \leq j$ . In this case,  $k$  is a candidate for  $i$ , iff  $N^{\mathbf{X}}(i + 1, k) < N^{\mathbf{X}}(i, k)$  and for all  $i < k' < k$ ,  $1 + N^{\mathbf{X}}(i + 1, k' - 1) + N^{\mathbf{X}}(k' + 1, k) < N^{\mathbf{X}}(i, k)$ . Operationally, during the computation of  $N^{\mathbf{X}}(i, k)$  we detect that  $k$  is a candidate for  $i$  by checking that the instance  $1 + N^{\mathbf{X}}(i + 1, k - 1) + N^{\mathbf{X}}(k + 1, k)$  of recursion case Eq. 2(b) is the only maximal case.

For non-candidates  $k$  there exists some  $k'$ ,  $i \leq k' < k$ , where  $N^{\mathbf{X}}(i, k) = N^{\mathbf{X}}(i, k') + N^{\mathbf{X}}(k' + 1, k)$ . Then for all  $j > k$ ,  $N^{\mathbf{X}}(i, k) + N^{\mathbf{X}}(k + 1, j) = N^{\mathbf{X}}(i, k') + N^{\mathbf{X}}(k' + 1, k) + N^{\mathbf{X}}(k + 1, j)$ , and by triangle inequality  $N^{\mathbf{X}}(i, k) + N^{\mathbf{X}}(k + 1, j) \leq N^{\mathbf{X}}(i, k') + N^{\mathbf{X}}(k' + 1, j)$ . This means that, whenever a non-candidate  $k$  yields a maximal value, then there is already a  $k' < k$  that yields the same value. Therefore  $k$  does not need to be considered, because the smallest such  $k'$  is taken into account.

Wexler et al. showed that sparsification reduces the expected time complexity of RNA folding by a linear factor, since the expected number of candidates for each  $i$  is constant. The transfer of sparsification to cases Eq 1(d) and (e) is straightforward, because only one subsequence is decomposed and the indices of the other subsequence remain fixed.

*Sparsifying recursion cases for joint structure folding.* We extend the sparsification idea to the recursion cases Eq 1(f) and (g), which split both sequences and therefore minimize over a pair of split points  $(k_R, k_S)$ . For the four dimensional matrix  $N(i_R, j_R, i_S, j_S)$ , the following generalization of the triangle inequality holds.

**Observation 1 (Triangle inequality for  $N(i_R, j_R, i_S, j_S)$ ).** For every subsequence  $[i_R, j_R]$  and  $[i_S, j_S]$  and for every  $i_R < k_R \leq j_R$  and  $i_S \leq k_S < j_S$ ,  $N(i_R, j_R, i_S, j_S) \geq N(i_R, k_R, i_S, k_S) + N(k_R + 1, j_R, k_S + 1, j_S)$ .

Note that in principle both cases Eq 1(f) and (g) split the two subsequences at  $k_R$  and  $k_S$ , respectively, into the pairs  $[i_R, k_R]$ ,  $[i_S, k_S]$  and  $[k_R + 1, j_R]$ ,  $[k_S + 1, j_S]$ .

The only difference is that within the first pair of subsequences,  $[i_R, k_R], [i_S, k_S]$ , case (f) assumes an arc  $i_R \bullet k_R$  and case (g) assumes an arc  $i_S \bullet k_S$ . We consider only the case Eq. **1(f)**, the case (g) is analogous.

**Definition 1 (Candidate for case Eq. **1(f)**).** For case Eq. **1(f)**, a pair  $(k_R, k_S)$  is a candidate for  $(i_R, i_S)$ , iff  $i_R$  and  $k_R$  are complementary and for all  $(k'_R, k'_S) \neq (k_R, k_S)$  with  $i_R < k'_R \leq k_R, i_S < k'_S \leq k_S$ ,

$$1 + N(i_R + 1, k_R - 1, i_S, k_S) + N(k_R + 1, k_R, k_S + 1, k_S) > 1 + N(i_R + 1, k'_R - 1, k'_S, k_S) + N(k'_R + 1, k_R, k'_S + 1, k_S),$$

With respect to the recursion case (f) a candidate  $(k_R, k_S)$  implies that the instance with  $k_R = j_R$  and  $k_S = j_S$  (i.e.  $1 + N(i_R + 1, k_R - 1, i_S, k_S) + N(k_R + 1, k_R, k_S + 1, k_S)$ ) is the only maximal instance in the maximization of (f). Furthermore, it implies that none of the cases (a)-(e) in the computation of  $N(i_R, k_R, i_S, k_S)$  yields a larger value than case (f).

**Lemma 1.** For correctness of the recursion of Eq. **1** in the maximization of Eq. **1(f)** it suffices to consider only the set of candidates given above.

*Proof.* For any non-candidate  $(k_R, k_S)$ , there exists some  $(k'_R, k'_S)$  with  $i_R - 1 \leq k'_R \leq k_R, i_S - 1 \leq k'_S \leq k_S, (k'_R, k'_S) \neq (k_R, k_S), (k'_R, k'_S) \neq (i_R - 1, i_S - 1)$ , and

$$1 + N(i_R + 1, k_R - 1, i_S, k_S) \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, k_R, k'_S + 1, k_S). \tag{3}$$

Note that  $k'_R = i_R - 1$  or  $k'_S = i_S - 1$  in Eq. **3** occurs when  $(k_R, k_S)$  is not a candidate due to one of the recursion cases (a)-(e).

Eq. **3** and the triangle inequality imply that for all  $j_R > k_R$  and  $j_S > k_S$

$$\begin{aligned} & 1 + N(i_R + 1, k_R - 1, k_S, j_S) + N(k_R + 1, j_R, k_S + 1, j_S) \\ & \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, k_R, k'_S + 1, k_S) + N(k_R + 1, j_R, k_S + 1, j_S) \\ & \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, j_R, k'_S + 1, j_S). \end{aligned} \tag{4}$$

Non-candidates  $(k_R, k_S)$  for  $(i_R, i_S)$  do not need to be considered in the recursions of all  $N(i_R, j_R, i_S, j_S)$ , because there exists a recursion case splitting at  $(k'_R, k'_S)$  that yields the same or better score for  $N(i_R, k_R, i_S, k_S)$ . The equivalent case is considered in the recursion of  $N(i_R, j_R, i_S, j_S)$  and, due to Eq. **4**, yields a greater or equal score.  $\square$

Therefore the recursion case Eq. **1(f)** can be updated such that the maximization runs only over the candidates for this case.

$$\max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ candidate for } (i_R, i_S)}} \left( 1 + N(i_R + 1, k_R - 1, i_S, k_S) + N(k_R + 1, j_R, k_S + 1, j_S) \right) \tag{5}$$

Analogously, we define candidates for case Eq. **1(g)**. The candidate criterion for Eq. **1(g)** is stricter than for Eq. **1(f)**, since we require that a candidate for Eq. **1(g)** is better than all cases Eq. **1(a)**-(e) and (f).

**Definition 2 (Expected number of candidates).**  $\psi_1(n)$  denotes the expected number of candidates  $k \leq n + i$  for some  $i$  in cases Eq. 1(d),(e), and Eq. 2(b).  $\psi_2(n)$  is the expected number of candidates  $(k_R, k_S)$ ,  $k_R \leq i_R + n$ ,  $k_S \leq i_S + n$ , for some  $(i_R, i_S)$  in cases Eq. 1(f) and (g).

Applying the described sparsification to all non-constant cases in recursions Eq. 1 and Eq. 2, yields the following.

**Theorem 2.**  $N(1, L_R, 1, L_S)$  can be computed in  $O((\psi_1(n) + \psi_2(n))n^4)$  expected time, where  $n = \max(L_R, L_S)$ .

For a theoretical bound on  $\psi_1(n)$  and  $\psi_2(n)$ , we assume the polymer-zeta property holds for each one of the RNA sequences that are involved in the interaction (with the other RNA sequence). The polymer-zeta property states that in any long polymer chain the probability of having arc between two monomers with distance  $m$  converges to  $b \cdot m^{-c}$ , where  $b, c > 0$  are some constants. For a polymer as a self-avoiding random walk on a square lattice, it has been known that  $c > 1$  [35]. The exponent  $c$  for the denaturation transition of DNA in both 2D and 3D models is found to be larger than 2 [36]. Since RNA folds similar to other polymers, one can assume that RNA folding obeys the polymer-zeta property; i.e. the probability that a structure is formed over the subsequence of length  $m$  converges to  $b \cdot m^{-c}$ , where  $c > 1$ . Although the property is not proven for RNA molecules, there is empirical evidence, as shown by Wexler et al. [18], that a version of polymer-zeta property holds for RNA molecules as well.

**Lemma 2.** Assume that the two interacting RNAs independently satisfy the polymer-zeta property with  $c > 1$ , i.e. there exist constants  $b > 0$  and  $c > 1$  such that the probability for any internal base pair  $i \bullet (i + m)$  is bounded by  $b \cdot m^{-c}$  - even when two RNAs interact. Then  $\psi_1(n) = O(1)$  and  $\psi_2(n) = O(n)$ .

*Proof.*  $\psi_1(n) = O(1)$  follows from Wexler et al. [18]. For  $\psi_2(n) = O(n)$ , consider all candidates  $(k_R, k_S)$  for  $(i_R, i_S)$  and case Eq. 1(f). (Case Eq. 1(g) is symmetric.) Note that in Eq. 1(f),  $i_R \bullet k_R$ . For a fixed  $k_S$  analogously to Wexler et al. [18], the expected number of  $k_R$  with  $i_R \bullet k_R$  is  $b \sum_{i=1}^n i^{-c} < b \sum_{i=1}^{\infty} i^{-c}$  which converges to a constant for  $c > 1$ . Hence for each of the  $O(n)$  possible values of  $k_S$ ,  $k_R$  takes only a constant number of different values and hence on average we have  $O(n)$  such candidates.  $\square$

**Space efficient strategy.** The space complexity of the algorithm can be reduced from  $O(n^4)$  to  $O(n^3 + \psi(n)n^2)$  as follows. The matrices  $N^R$  and  $N^S$  only require  $O(n^2)$  space. All cases for the computation of an entry  $N(i_R, j_R, i_S, j_S)$  only rely on entries  $N(i'_R, j'_R, i'_S, j'_S)$  that satisfy one of the following two properties. (i)  $j'_R \in \{j_R - 1, j_R\}$  and  $j'_S \in \{j_S - 1, j_S\}$  or (ii)  $N(i'_R, j'_R, i'_S, j'_S)$  corresponds to some candidate of the respective case, i.e. in case Eq. 1(d)  $j'_R + 1$  is a candidate for  $i'_R - 1 = i_R$ , in case (e)  $j'_S + 1$  is a candidate for  $i'_S - 1 = i_S$ , in case (f)  $(j'_R + 1, j'_S)$  is a candidate for  $(i'_R - 1, i'_S) = (i_R, j_R)$ , and in case (g)  $(j'_R, j'_S + 1)$  is a candidate for  $(i'_R, i'_S - 1) = (i_R, j_R)$ . As shown in the following

**Algorithm:** Space efficient evaluation of Eq. [1](#)

```

precompute matrices  $N^{\mathbf{R}}$  and  $N^{\mathbf{S}}$  ;
initialize empty lists for candidates ;
for  $j_R = 1..L_R$  do
  allocate and init matrix slice  $N(\cdot, j_R, \cdot, \cdot)$  ;
  for  $j_S = 1..L_S, i_R = j_R..1, i_S = j_S..1$  do
    compute  $N(i_R, j_R, i_S, j_S)$  ;
    if  $j_R$  is candidate for  $i_R$  and Eq. 1\(d\) then
      store  $N^{\mathbf{R}}(i_R + 1, j_R - 1, i_S, j_S)$  in list for  $i_R$  and Eq. 1\(d\)
    else if  $j_S$  is candidate for  $i_S$  and Eq. 1\(e\) then
      store  $N^{\mathbf{S}}(i_S + 1, j_S - 1)$  in list for  $i_S$  and Eq. 1\(e\)
    else if candidate for Eq. 1\(f\) then
      store  $N(i_R + 1, j_R - 1, i_S, j_S)$  in list for  $(i_R, i_S)$  and Eq. 1\(f\)
    else if candidate for Eq. 1\(g\) then
      store  $N(i_R, j_R, i_S + 1, j_S + 1)$  in list for  $(i_R, i_S)$  and Eq. 1\(g\)
    end
  end
  free matrix slice  $N(\cdot, j_R - 1, \cdot, \cdot)$  ;
end

```

algorithm, all values that satisfy (i) can be stored in a three dimensional matrix and all values that satisfy (ii) can be stored in candidate lists of length  $\psi(n)$  for each of the  $O(n^2)$  instances of  $(i_R, i_S)$ .

Note that, in the pseudocode, we maintain two three dimensional matrices, namely  $N(\cdot, j_R, \cdot, \cdot)$  and  $N(\cdot, j_R - 1, \cdot, \cdot)$  during the computation of the values for  $j_R$ . In practice, we save half of this memory, because any entry  $N(\cdot, j_R - 1, \cdot, j_S)$  can be freed as soon as all  $N(\cdot, j_R, \cdot, j_S)$  are computed.

*Trace-Back.* We describe the recursive trace-back starting from a matrix entry  $(i_R, j_R, i_S, j_S)$ . Computing the Trace-back involves some recomputation. First, the entire matrix slice  $N(\cdot, j_R, \cdot, j_S)$  is recomputed unless it is already in memory. This requires access to only entries in the same matrix slice and candidates. Then, the best case in the recursion for  $N(i_R, j_R, i_S, j_S)$  is identified. In cases (a)-(c), we recurse to the respective entry. In cases (d)-(g), which split in a first and second entry, we first recurse to the second one, which is in the same matrix slice. Then, we free the memory for the current matrix slice and recurse to the first entry, which will cause recomputation. Since each entry is recomputed at most once, the trace-back does not affect the asymptotic complexity.

### 3.2 Sparsification for Minimizing Free Energy

Alkan et al. [1](#) describe minimization of the free energy of RNA-RNA-interaction based on a simple stacked-pair energy model assuming there are no pseudoknots, crossing bonds, and zigzags in the joint structure. Here we discuss an algorithm for RNA-RNA interaction free energy minimization on the same type of interactions based on the interaction energy model of Chitsaz et al. [2](#). Since the general recursive structure of this algorithm is identical to base pair maximization, our sparsification technique can be applied to reduce their time and space

complexity in the same way. The exact recursions of our sparsified free energy minimization algorithm are given in the appendix. Compared to base pair maximization, these recursions distinguish several matrices representing differently scored substructures. Notably, they are formulated such that all cases that split an entry  $(i_R, j_R, i_S, j_S)$  at  $(k_R, k_S)$  are of the same form as cases Eq. 1(f) and (g) or  $k_R$  and  $k_S$  are bounded due to the loop length restriction of the energy model. Achieving the same space complexity requires one additional consideration. For assigning correct energy to internal loops formed by interaction arcs, an entry  $(i_R, j_R, i_S, j_S)$  can depend on  $(i'_R, j'_R, i_S, j_S)$ , where  $j'_R$  is neither  $j_R$  nor  $j_R - 1$ . However,  $j_R - j'_R$  is still bounded by the maximal loop length  $\ell$  of the energy model, i.e.  $j_R - j'_R < \ell$ . Hence, it suffices to store  $\ell$  matrix slices  $(\cdot, j'_R, \cdot, \cdot)$  for  $j_R - \ell < j'_R \leq j_R$ .

**Theorem 3.** *The MFE interaction of two RNAs of maximal length  $n$  can be computed in expected time  $O((\psi_1(n) + \psi_2(n))n^4)$  and expected space  $O((\psi_1(n) + \psi_2(n))n^2 + n^3)$ .*

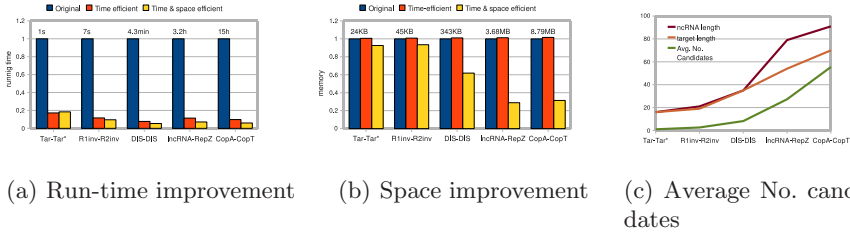
## 4 Experimental Results

For evaluating the effect of sparsification on RNA-RNA-interaction, we implemented three variants of the total free energy minimization algorithm for RNA-RNA-interaction prediction: the first variant does not perform any sparsification, the second employs sparsification for improving the time complexity, and the third improves both time and space complexity. Below, we first demonstrate that sparsification leads to a significant reduction of the time and space requirements in practice. Then we study the relationship between the sequence length and the number of candidates per each base on a large set of confirmed RNA-RNA interactions and study the average time/space behavior of the algorithms.

Since sparsification does not affect the calculated free energy values (i.e. optimality of the calculated joint free energy of the interaction), the accuracy of the predicted interactions is identical to previous approaches for general RNA-RNA-interactions based on the same scoring scheme [12,32]. As a result, the reader is referred to Salari et al. [32] for an assessment of sensitivity, positive prediction value, and F-measure of these methods (which will be identical to that of the method presented here) on the data set of Kato et al. [37] which involves five distinct RNA-RNA interactions.

### 4.1 Time and Space Requirements of Total Free Energy Minimization

We applied the three variants of the MFE algorithm to five distinct RNA-RNA interactions reported by Kato et al. [37], which were used by Salari et al. [32] to assess the accuracy of available RNA-RNA interaction methods with no sparsification. Note that the available methods are not capable of handling interactions involving longer RNAs.



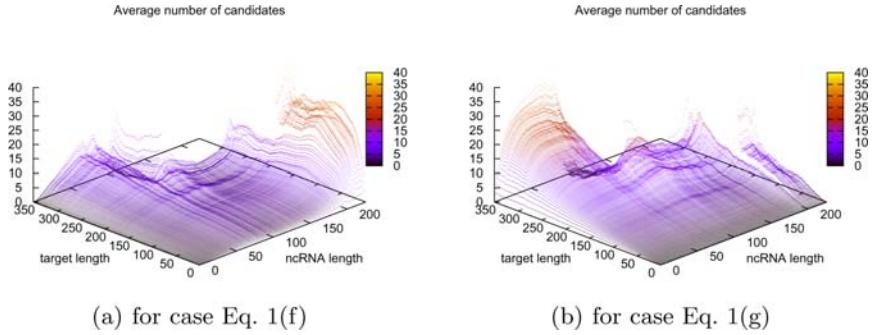
**Fig. 2.** Performance of three variants of the RNA-RNA interaction prediction algorithm via total free energy minimization, on a set of interactions compiled by Kato et al. [37]. All values for time and space usage are normalized by the usage of the non-sparsified algorithm, for which absolute time/space usage figures are also given.

Fig. 2 shows (in absolute terms) time and space usage of the algorithms (with or without sparsification) on a Sun Fire X4600 server with 2.6 GHz processor speed. The results show that sparsification significantly improves the performance of the algorithms. In fact, Fig. 2 demonstrates that as the RNA sequences in question get longer, the relative performance of the sparsified algorithms (with respect to the non-sparsified ones) improve. Although the pure time optimization causes a small space overhead due to maintaining the candidate lists, the time and space optimization not only improve the space utilization, as expected, but also results in further reduction in running time.

## 4.2 Number of Candidates

The time and space complexity of the (time and space) sparsified RNA-RNA-interaction prediction algorithm is linearly proportional to the (average) number of interaction partner candidates per base. Fig. 2(c) shows how the average number of candidates ( $k_R$ ,  $k_S$ ) change as the lengths of the two RNA sequences increase. While the non-sparsified algorithms need to consider a quadratic number of split points ( $k_R$ ,  $k_S$ ), the number of candidates (and hence the number of split points) is much lower for the sparsified algorithms.

In order to observe the effects of sparsification on a much larger data set involving longer RNA sequences, we employ the algorithm for RNA-RNA interaction prediction which maximizes the number of (internal and external) base pairs. The data set we use for this purpose includes 43 pairs of ncRNAs and their known target mRNAs. This set not only includes (i) the data set of Kato et al. [37], but also (ii) a recently compiled test set of Busch et al. [30] consisting of 18 sRNA-target pairs, as well as (iii) all ncRNA-target interactions of E.coli from NPinter [38]. Among these interactions 32 are from E.coli, 8 are from Salmonella typhimurium and 3 are from HIV. Since the majority of the known ncRNAs bind to their target mRNAs in close proximity of the start codon, we extracted - as the target region - a subsequence comprising 300nt upstream and 50nt downstream of the first base of the start codon of each mRNA from



**Fig. 3.** Average number of candidates as a function of subsequence lengths

GenBank [39]. As a result, the maximum sequence length is 227nt for ncRNAs and 350nt for target mRNAs.

The experimental results on this larger data set confirm that the sparsification technique works for a single RNA folding via base pair maximization: the average number of candidates for those cases is low (roughly 5) as previously reported by Wexler et al. [18].

The recursion cases Eq. II(f) and (g) split both RNAs simultaneously at points  $(k_R, k_S)$ . Therefore they dominate the running time of the algorithm. For these cases, we counted the candidates that were considered during the computation of (the maximum number of base pairs of) each subsequence pair. The average number of candidates for different subsequence lengths, both for ncRNAs and mRNAs are depicted in Fig. 3 - specific cases that correspond to Eq. II(f) as well as Eq. II(g) are provided separately. Note that the average number of candidates are generally low regardless of the sequence lengths: among all possible combinations of split points  $(k_R, k_S)$  (respectively in ncRNA and mRNA), even for the longest subsequences (e.g. ncRNA length  $l_S = 252$  and mRNA length  $l_R = 202$ ), no more than 40 pairs (of the possible  $252 \times 202 = 50,904$  combinations for this example) are actual candidates on the average<sup>2</sup>.

## 5 Conclusion

In this paper, we consider the problem of predicting the joint structure of two interacting RNAs via minimizing their total free energy as a tool for detecting/verifying mRNA targets of regulatory ncRNAs. Earlier approaches to the problem either use a restricted interaction model, not covering many known joint structures, or require significant computational resources for many practical instances. Here we show that sparsification, a technique that has been applied

<sup>2</sup> Note that certain combinations of  $l_R$  and  $l_S$  there is no value for the number of candidates due to the fact that there is no data for  $l_R > 111$  and  $l_S > 252$  as well as  $l_R > 202$  and  $l_S > 153$ .



to single RNA folding, can be applied to the problem of RNA-RNA interaction prediction, to significantly improve both the running time and the space utilization of these approaches. In fact, by employing a version of the polymer-zeta property for interacting RNA-structures (a property generally assumed to be held by many polymers, and has been empirically shown for single RNAs), we show how to reduce the running time and space of RNA-RNA interaction prediction, from  $O(n^6)$  time and  $O(n^4)$  space to  $O(n^4\psi(n))$  time and  $O(n^2\psi(n) + n^3)$  space, for a function  $\psi(n) = O(n)$  on average. These theoretical predictions are verified by our experiments; as a result it is now possible to employ computational prediction of RNA-RNA interactions to a much wider range of potential regulatory ncRNAs and their targets.

## Acknowledgments

R. Salari was supported by SFU-CTEF funded Bioinformatics for Combating Infectious Diseases Project co-lead by Sahinalp. The research of M. Möhl was funded by the German Research Foundation (DFG grant BA 2168/3-1). S.C. Sahinalp was supported by MITACS, NSERC, the CRC program and the Michael Smith Foundation for Health Research. R. Backofen received funding from the German Research Foundation (DFG grant BA 2168/2-1 SPP 1258), and from the German Federal Ministry of Education and Research (BMBF grant 0313921 FRISYS).

## References

1. Alkan, C., Karakoc, E., Nadeau, J.H., Sahinalp, S.C., Zhang, K.: RNA-RNA interaction prediction and antisense RNA target search. *Journal of Computational Biology (Special RECOMB 2005 Issue)* 13(2), 267–282 (2006)
2. Chitsaz, H., Salari, R., Sahinalp, S.C., Backofen, R.: A partition function algorithm for interacting nucleic acid strands. *Bioinformatics (Special ISMB/ECCB 2009 Issue)* 25(12), i365–i373 (2009)
3. Storz, G.: An expanding universe of noncoding RNAs. *Science* 296(5571), 1260–1263 (2002)
4. Bartel, D.P.: MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell* 116(2), 281–297 (2004)
5. Hannon, G.J.: RNA interference. *Nature* 418(6894), 244–251 (2002)
6. Zamore, P.D., Haley, B.: Ribo-gnome: the big world of small RNAs. *Science* 309(5740), 1519–1524 (2005)
7. Wagner, E., Flardh, K.: Antisense RNAs everywhere?. *Trends Genet.* 18, 223–226 (2002)
8. Brantl, S.: Antisense-RNA regulation and RNA interference. *Bioch. Biophys. Acta* 1575(1-3), 15–25 (2002)
9. Gottesman, S.: Micros for microbes: non-coding regulatory RNAs in bacteria. *Trends in Genetics* 21(7), 399–404 (2005)
10. Seeman, N.: From genes to machines: DNA nanomechanical devices. *Trends Biochem. Sci.* 30, 119–125 (2005)



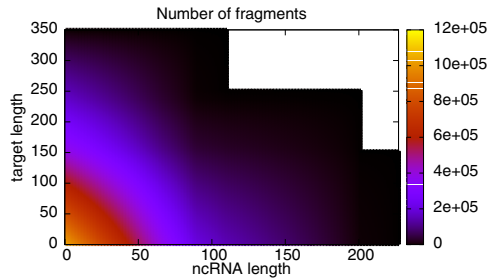
11. Seeman, N.C., Lukeman, P.S.: Nucleic acid nanostructures: bottom-up control of geometry on the nanoscale. *Reports on Progress in Physics* 68, 237–270 (2005)
12. Simmel, F., Dittmer, W.: DNA nanodevices. *Small* 1, 284–299 (2005)
13. Venkataraman, S., Dirks, R., Rothmund, P., Winfree, E., Pierce, N.: An autonomous polymerization motor powered by DNA hybridization. *Nat. Nanotechnol.* 2, 490–494 (2007)
14. Yin, P., Hariadi, R., Sahu, S., Choi, H., Park, S., Labean, T., Reif, J.: Programming DNA tube circumferences. *Science* 321, 824–826 (2008)
15. Pervouchine, D.D.: IRIS: intermolecular RNA interaction search. *Genome Inform.* 15(2), 92–101 (2004)
16. Huang, F.W., Qin, J., Reidys, C.M., Stadler, P.F.: Partition Function and Base Pairing Probabilities for RNA-RNA Interaction Prediction. *Bioinformatics* 25(20), 2646–2654 (2009)
17. David, L., Huber, W., Granovskaia, M., Toedling, J., Palm, C.J., Bofkin, L., Jones, T., Davis, R.W., Steinmetz, L.M.: A high-resolution map of transcription in the yeast genome. *Proc. Natl. Acad. Sci. U.S.A.* 103(14), 5320–5325 (2006)
18. Wexler, Y., Zilberstein, C., Ziv-Ukelson, M.: A study of accessible motifs and RNA folding complexity. *Journal of Computational Biology (Special RECOMB 2006 Issue)* 14(6), 856–872 (2007)
19. Backofen, R., Tsur, D., Zakov, S., Ziv-Ukelson, M.: Sparse RNA folding: Time and space efficient algorithms. In: Kucherov, G., Ukkonen, E. (eds.) *CPM 2009. LNCS*, vol. 5577, pp. 249–262. Springer, Heidelberg (2009)
20. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.* 45(5), 810–825 (1985)
21. Ziv-Ukelson, M., Gat-Viks, I., Wexler, Y., Shamir, R.: A faster algorithm for RNA co-folding. In: Crandall, K.A., Lagergren, J. (eds.) *WABI 2008. LNCS (LNBI)*, vol. 5251, pp. 174–185. Springer, Heidelberg (2008)
22. Rehmsmeier, M., Steffen, P., Höchsmann, M., Giegerich, R.: Fast and effective prediction of microRNA/target duplexes. *RNA* 10(10), 1507–1517 (2004)
23. Tjaden, B., Goodwin, S.S., Opdyke, J.A., Guillier, M., Fu, D.X., Gottesman, S., Storz, G.: Target prediction for small, noncoding RNAs in bacteria. *Nucleic Acids Research* 34(9), 2791–2802 (2006)
24. Andronescu, M., Zhang, Z.C., Condon, A.: Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology* 345(5), 987–1001 (2005)
25. Bernhart, S.H., Tafer, H., Mückstein, U., Flamm, C., Stadler, P.F., Hofacker, I.L.: Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol. Biol.* 1(1), 3 (2006)
26. Dirks, R.M., Bois, J.S., Schaeffer, J.M., Winfree, E., Pierce, N.A.: Thermodynamic analysis of interacting nucleic acid strands. *SIAM Review* 49(1), 65–88 (2007)
27. Zuker, M.: Prediction of RNA secondary structure by energy minimization. *Methods in Molecular Biology* 25, 267–294 (1994)
28. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. *Monatshefte Chemie* 125, 167–188 (1994)
29. Mückstein, U., Tafer, H., Hackermüller, J., Bernhart, S.H., Stadler, P.F., Hofacker, I.L.: Thermodynamics of RNA-RNA binding. *Bioinformatics* 22(10), 1177–1182 (2006)
30. Busch, A., Richter, A.S., Backofen, R.: IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics* 24(24), 2849–2856 (2008)

31. Argaman, L., Altuvia, S.: fhla repression by OxyS RNA: kissing complex formation at two sites results in a stable antisense-target RNA complex. *Journal of Molecular Biology* 300(5), 1101–1112 (2000)
32. Salari, R., Backofen, R., Sahinalp, S.C.: Fast prediction of RNA-RNA interaction. In: Salzberg, S.L., Warnow, T. (eds.) WABI 2009. LNCS, vol. 5724, pp. 261–272. Springer, Heidelberg (2009); Also Algorithms for Molecular Biology (in press)
33. Chitsaz, H., Backofen, R., Sahinalp, S.C.: biRNA: Fast RNA-RNA binding sites prediction. In: Salzberg, S.L., Warnow, T. (eds.) WABI 2009. LNCS, vol. 5724, pp. 25–36. Springer, Heidelberg (2009)
34. Nussinov, R., Pieczenik, G., Griggs, J.R., Kleitman, D.J.: Algorithms for loop matchings. *SIAM Journal on Applied Mathematics* 35(1), 68–82 (1978)
35. Fisher, M.E.: Shape of a self-avoiding walk or polymer chain. *Journal of Chemical Physics* 44, 616–622 (1966)
36. Kafri, Y., Mukamel, D., Peliti, L.: Why is the DNA denaturation transition first order? *Phys. Rev. Lett.* 85, 4988–4991 (2000)
37. Kato, Y., Akutsu, T., Seki, H.: A grammatical approach to rna-rna interaction prediction. *Pattern Recogn.* 42(4), 531–538 (2009)
38. Wu, T., Wang, J., Liu, C., Zhang, Y., Shi, B., Zhu, X., Zhang, Z., Skogerb, G., Chen, L., Lu, H., Zhao, Y., Chen, R.: NPInter: the noncoding RNAs and protein related biomacromolecules interaction database. *Nucleic Acids Res.* 34, D150–D152 (2006)
39. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L.: GenBank. *Nucleic Acids Research* 36(Database issue), D25–D30 (2008)

## Appendix

*Total number of fragments for different ncRNA and target subsequence lengths.* The plot of Fig. 4 shows the total number of fragments for different ncRNA and target subsequence lengths. The white region on top right of the plot in Fig. 4 ( $l_R > 111 \wedge l_S > 252$  and  $l_R > 202 \wedge l_S > 153$ ) denotes the area that there are no fragments in our data set.

*Sparsification of Energy Minimization RNA-RNA-Interaction.* Here, we present our sparsified algorithm for RNA-RNA interaction free energy minimization based on the interaction energy model of Chitsaz et al. [2]. The minimum free



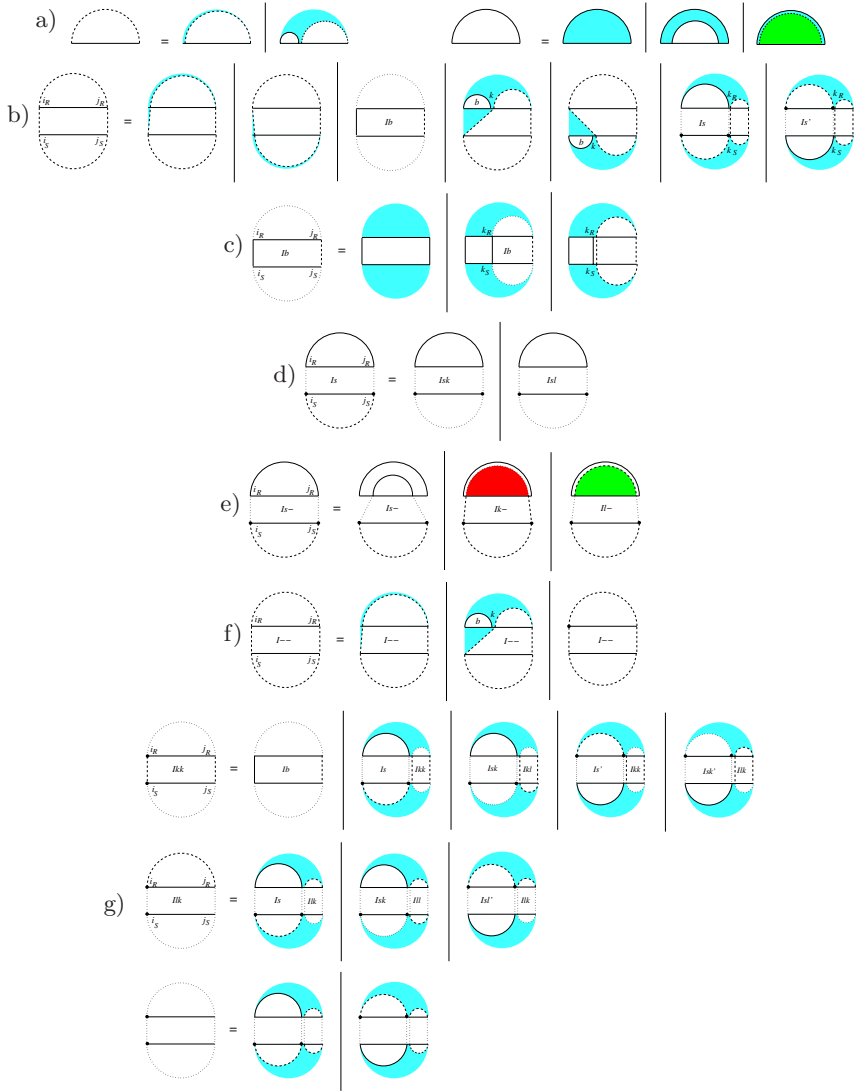
**Fig. 4.** Total number of fragments for different ncRNA and target lengths

energy (mfe) joint structure  $M(i_R, j_R, i_S, j_S)$  derived from one of the seven possible cases shown in Fig. 5(b). The first two cases are when  $i_R$  or  $i_S$  is an unpaired base. In third case  $i_R$  interacts with  $i_S$ , this bond starts a special type of joint structure denoted by  $Ib$  and it is explained in Fig. 5(c). The fourth and fifth cases are when  $i_R$  or  $i_S$  is forming intramolecular base pairs. In other possible cases either  $i_R \bullet k_R$  is an interaction arc subsuming  $[i_S, k_S]$  or  $i_S \bullet k_S$  is an interaction arc subsuming  $[i_R, k_R]$ . The sparsified DP algorithm for free energy minimization,  $M(i_R, j_R, i_S, j_S)$ , is defined as follows:

$$M(i_R, j_R, i_S, j_S) = \max \left\{ \begin{array}{ll} M(i_R + 1, j_R, i_S, j_S) & (a) \\ M(i_R, j_R, i_S + 1, j_S) & (b) \\ M^{Ib}(i_R, j_R, i_S, j_S) & (c) \\ \max_{\substack{i_R < k \leq j_R \\ k \text{ cand. for } (i_R)}} \left( \begin{array}{l} M^{\mathbf{R}.b}(i_R, k) \\ + M(k + 1, j_R, i_S, j_S) \end{array} \right) & (d) \\ \max_{\substack{i_S \leq k < j_S \\ k \text{ cand. for } (i_S)}} \left( \begin{array}{l} M^{\mathbf{S}.b}(i_S, k) \\ + M(i_R, j_R, k + 1, j_S) \end{array} \right) & (e) \\ \max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ cand. for } (i_R, i_S)}} \left( \begin{array}{l} M^{Is}(i_R, k_R, i_S, k_S) \\ + M(k_R + 1, j_R, k_S + 1, j_S) \end{array} \right) & (f) \\ \max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ cand. for } (i_R, i_S)}} \left( \begin{array}{l} M^{Is'}(i_R, k_R, i_S, k_S) \\ + M(k_R + 1, j_R, k_S + 1, j_S) \end{array} \right) & (g) \end{array} \right. \quad (6)$$

$$M^{\mathbf{X}}(i, j) = \max \left\{ \begin{array}{ll} M^{\mathbf{X}}(i + 1, j) & (a) \\ \max_{\substack{i < k \leq j \\ (k) \text{ cand. for } (i)}} \left( \begin{array}{l} M^{\mathbf{X}.b}(i, k) \\ + M^{\mathbf{X}}(k + 1, j) \end{array} \right) & (b) \end{array} \right. \quad (7)$$

$M^{Ib}(i_R, j_R, i_S, j_S)$  (Fig. 5(c)) is the mfe for the joint structure of  $[i_R, j_R]$  and  $[i_S, j_S]$  assuming  $i_R \cdot j_S$  is an interaction bond, and  $M^{Is}(i_R, j_R, i_S, j_S)$  (Fig. 5(d)) is the mfe for the joint structure of  $[i_R, j_R]$  and  $[i_S, j_S]$  assuming  $i_R \circ j_R$  is an interaction arc subsuming  $[i_S, j_S]$ .  $M^{Is'}$  is symmetric to  $M^{Is}$  where  $i_S \circ j_S$  is an interaction arc subsuming  $[i_R, j_R]$ . In  $Q^{Isl}$ ,  $[i_S, j_S]$  contains at least interaction arc and in  $Q^{Isk}$ ,  $[i_S, j_S]$  contains at least one direct bond. The other auxiliary matrices are  $Q^{Ill}$ ,  $Q^{Ilk}$ ,  $Q^{Ikl}$ , and  $Q^{Ikk}$  (Fig. 5(g)).  $Q^{Ill}$  includes all cases where both  $[i_R, j_R]$  and  $[i_S, j_S]$  have at least one interaction arc.  $Q^{Ilk}$  (symmetric to  $Ikl$ ) includes all cases where  $[i_R, j_R]$  has at least one interaction arc and  $[i_S, j_S]$  has at least one direct bond.  $Q^{Ikk}$  includes all cases where both  $[i_R, j_R]$  and  $[i_S, j_S]$  have at least one direct bond.



**Fig. 5.** a) Recursion cases for MFE single structure. b) Recursion cases for MFE joint structure. c) Recursion cases for MFE joint structure while  $i_R \circ j_S$  is a bond. Here  $i_R < k_R \leq \min i_R + \ell, j_R$  and  $i_S < k_S \leq \min i_S + \ell, j_S$  w.  $\ell$  is the maximal loop length. d) In recursive quantity  $Is$ ,  $i_R \bullet j_R$  is an interaction arc which subsumes interval  $[i_S, j_S]$ . The subsumed area contains at least one direct bond or at least one interaction arcs. e) Recursion cases for  $Isl$  or  $Isk$  which extract the interaction arc  $i_R \bullet j_R$ . f) In  $Ikk$ ,  $Ikl$ ,  $Ilk$ , or  $Ill$ , if the terminal point  $i_R$  (or  $j_S$ ) is not an end point of interaction bond or arc, some recursions should be applied to extract the internal structure. g) Recursion for joint structures that has direct interactions on both subsequences ( $Ikk$ ), direct interaction on one subsequence and interaction arc on the other ( $Ikl$  and  $Ilk$  which are symmetric), and interaction arcs on both subsequences ( $Ill$ ).

# HLA Type Inference via Haplotypes Identical by Descent

Manu N. Setty, Alexander Gusev, and Itsik Pe'er

Dept of Computer Science, Columbia University, New York NY 10027 USA  
`itsik@cs.columbia.edu`

**Abstract.** The Human Leukocyte Antigen (HLA) genes play a major role in adaptive immune response and are used to differentiate self antigens from non self ones. HLA genes are hyper variable with nearly every locus harboring over a dozen alleles. This variation plays an important role in susceptibility to multiple autoimmune diseases and needs to be matched on for organ transplantation. Unfortunately, HLA typing by serological methods is time consuming and expensive compared to high throughput Single Nucleotide Polymorphism (SNP) data. We present a new computational method to infer per-locus HLA types using shared segments Identical By Descent (IBD), inferred from SNP genotype data. IBD information is modeled as graph where shared haplotypes are explored among clusters of individuals with known and unknown HLA types to identify the latter. We analyze performance of the method in a previously typed subset of the HapMap population, achieving accuracy of 96% in HLA-A, 94% in HLA-B, 95% in HLA-C, 77% in HLA-DR1, 93% in HLA-DQA1 and 90% in HLA-DQB1 genes. We compare our method to a tag SNP based approach and demonstrate higher sensitivity and specificity. Our method demonstrates the power of using shared haplotype segments for large-scale imputation at the HLA locus.

## 1 Introduction

The Human Leukocyte Antigen (HLA) region, located on chromosome 6p21, encodes genes for the Major Histocompatibility Complex (MHC) in humans. MHC are cell surface proteins which play an important role in adaptive immune response. These proteins form a complex with the antigenic peptides which is presented on the cell surface. This complex is recognized by the T-cell receptors to trigger the adaptive immune response by inducing the death of the cell and/or production of antibodies.

The HLA genes are classified into two main classes. Class I genes present peptides from within the cell and are recognized by the CD8+/cytotoxic T cells which kill the cells displaying the antigens. The Class I MHC genes are HLA-A, HLA-B, HLA-C. Class II genes present peptides from the intra cellular vacuoles and are recognized by the CD4+/helper T cells which trigger antibody production. The Class II genes are HLA-DP, HLA-DM, HLA-DOA, HLA-DOB, HLA-DQ and HLA-DR. HLA genes are also highly polymorphic. For example,



pairs of individuals in linear time. Here, we present a graph-based method that uses segments shared between HLA-typed and un-typed individuals to infer their putative HLA types. We provide theoretic description of the model and offer software implementation, a unique contribution to the geneticist user.

The paper is organized as follows: We define the framework and the problem in Sec. 2. Section 3 describes our algorithms for HLA imputation. The data used for analysis is explained in Sec. 4. The results and comparison to tag SNP method are presented in Sec. 5, followed by a summary discussion in Sec. 6.

## 2 Preliminaries

We define a model for inferring HLA types at individual loci for unphased data. We study one locus at a time and throughout the methods sections consider only the current locus. The results repeat such analysis for each locus separately along the HLA region. An individual  $v$  is associated with a pair of alleles  $(\alpha, \beta)$  at each HLA locus, representing the HLA types. We denote this by  $v(\alpha, \beta)$ . An individual with  $\alpha = \beta$  is homozygous. The input consists of a set of individuals with known HLA types and another set with unknown HLA types. The individuals in these two sets are referred to as *resolved* and *unresolved* individuals, respectively. Unphased IBID segments that are shared pair-wise across resolved and unresolved individuals are inferred using GERMLINE [4] and serve as a starting point for our analysis.

Formally, IBID is represented as an undirected graph called the *IBID-Graph*,  $G_{\text{IBID}}$ . The nodes  $V$  of  $G_{\text{IBID}}$  map to the individuals with genotypic data (resolved and unresolved) and the edges  $E$  represent the IBID shared segments. Ideally we would have  $G_{\text{IBID}}$  as input for HLA imputation, but in practice we may only assume the input to be a noisy version  $G_{\text{IBID}}^0$  of the true  $G_{\text{IBID}}$ .  $G_{\text{IBID}}^0$  has the same nodes, as  $G_{\text{IBID}}$  along with many of the same edges (true positives), but it also contains false positives (edges between nodes not related by IBID) and false negatives (missing edges between nodes related by IBID).

An edge in  $G_{\text{IBID}}^0$  between two nodes  $v(\alpha, \beta)$  and  $w(\gamma, \delta)$  is suggestive of the nodes sharing one or both the HLA types i.e., at least one of  $(\alpha = \gamma)$ ,  $(\beta = \gamma)$ ,  $(\alpha = \delta)$  or  $(\beta = \delta)$  is true. The edges which satisfy these criteria are termed *consistent*. Note that the converse does not hold: if two nodes share a common HLA type, it does not imply they are IBID because the same HLA allele can have multiple SNP-haplotype backgrounds. The HLA imputation problem is intuitively defined as follows:

Input:  $G_{\text{IBID}}^0(V, E^0)$  and a set of assigned type pairs  $(\alpha_{(r)}, \beta_{(r)})$  for all nodes  $r$  in a resolved subset  $R \subset V$ .

Output: Assignment of type pairs  $(\alpha_{(u)}, \beta_{(u)})$  for all unresolved nodes  $u \in V \setminus R$ .

Objective: Maximize the correctly assigned nodes.

As the objective is not defined in terms of the available data, we consider a surrogate optimization criterion. We seek an assignment which maximizes the consistent edges.

We propose an iterative approach for HLA imputation. While  $G_{\text{IBD}}^0(V, E^0)$  is used as the input for the first iteration, the IBD-Graph is adjusted in each iteration to maintain the consistency of edges. Formally, denote the IBD-Graph in the  $i^{\text{th}}$  iteration as  $G_{\text{IBD}}^i$ . We detect false positives and false negatives which are removed from and added to the edge set respectively to form  $E^i$ , the edge set in the  $i^{\text{th}}$  iteration. After adjusting the graph, possible HLA types and HLA type-pairs are inferred for unresolved nodes. Possible HLA types represent alleles of one of the chromosomes satisfying the constraints defined by  $G_{\text{IBD}}^i$  and HLA type-pairs represent alleles of both the chromosomes of the unresolved node.

$G_{\text{IBD}}^i$  is examined in triplets of nodes,  $T(r_1, r_2, u)$ , where  $r_1, r_2$  are resolved and  $u$  is unresolved and at least two of the edges  $(r_1, r_2)$ ,  $(r_1, u)$  and  $(r_2, u)$  are in  $E_i$ . The possible HLA types and type-pairs from all triplets containing  $u$  are combined based on a likelihood function to assign the most likely HLA types to the unresolved node. We expect a number of unresolved nodes to be resolved within each iteration. This information is then used in subsequent iterations to infer HLA types for the remaining ambiguous or unresolved nodes (Fig. 2)

## 2.1 Sources of Information

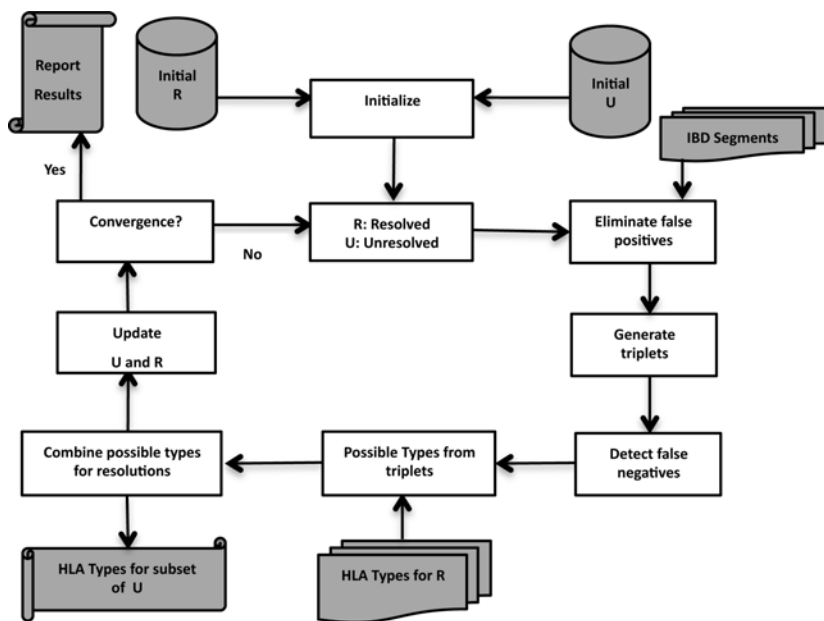
The sources of information for defining possible HLA types are triplets generated, matches with homozygote nodes and previously detected false negatives. Triplets and homozygote matches are deduced from  $G_{\text{IBD}}^i$ .

We define three possible configurations for a triplet based on the sub-graph of  $G_{\text{IBD}}^i$  induced by  $(r_1, r_2, u)$ . If this sub-graph is a clique, we call it a *triangle triplet* (Fig. 3a). Alternatively, it is a path along the three nodes and we denote this as an *end triplet* (Fig. 3b) or a *middle triplet* (Fig. 3c) depending on the position of  $u$  along the path. Possible HLA types are deduced from each triplet as described below.

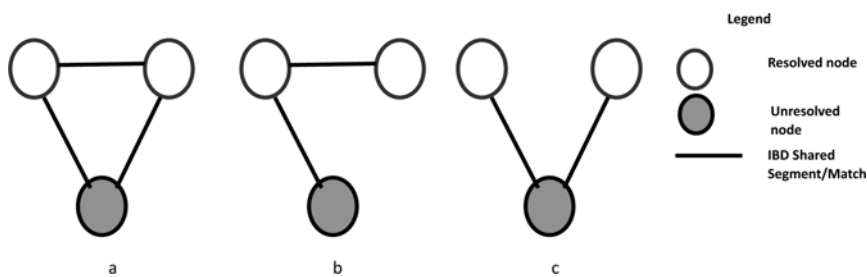
**Triangle triplets.** (Fig. 3a) are fully connected by definition. Since only consistent edges are considered for any triangle triplet  $T(r_1, r_2, u)$ ,  $r_1$  and  $r_2$  share one or both HLA types. We consider these cases in turn. (1) One shared HLA-type: We denote the HLA types of the resolved nodes by  $r_1(\rho, \alpha)$  and  $r_2(\rho, \beta)$  with  $\alpha \neq \beta$ . In this case the following assignments to  $u$  maintain the consistency of the edges: the shared HLA type:  $\rho$  or the HLA type-pair formed by  $(\alpha, \beta)$  (Fig. 4A(I)). (2) Both HLA types shared: Here, we denote the HLA types by  $r_1(\rho, \tau)$  and  $r_2(\rho, \tau)$ . Thus the shared types,  $\rho$  and  $\tau$ , are possible HLA types for  $u$  (Fig. 4A(II)).

**End triplets.** (Fig. 3b) are processed as follows: For a triplet  $T(r_1, r_2, u)$  assume without loss of generality that  $(r_1, u) \in E^i$  and  $(r_2, u) \notin E^i$ . We denote the HLA types of the resolved nodes by  $r_1(\rho, \alpha)$  and  $r_2(\rho, \beta)$ . By definition, if  $\alpha \neq \beta$ , then assigning the HLA type of  $r_1$  not shared with  $r_2$  i.e.;  $\alpha$ , to  $u$  maintains the consistency of the edges (Fig. 4B(I)). Otherwise if  $\alpha = \beta$ , the edge  $(r_1, r_2)$  is detected as a false negative and is added to  $E^i$ . The triplet is treated as a triangle triplet in the subsequent iterations. For example, the triplet in (Fig. 4B(II)) defines  $\rho$  and  $\alpha$  as possible HLA types.





**Fig. 2. An iterative-triangulation approach for HLA type inference from unphased data.** The method initializes the resolved and unresolved nodes from the training and test sets, respectively. The edges among these individuals are used to generate triplets. These triplets are used to draw up a set of possible HLA type resolutions for each node. The HLA types with highest likelihood are chosen as resolution for the nodes where applicable and the process is repeated for the remaining unresolved nodes.



**Fig. 3. Types of triplets** (a) Triangle triplet: Pair-wise matches between all individuals. (b) End triplet: Unresolved individual has match with only of the resolved individuals. (c). Middle triplet. Resolved individuals do not have a match.

**Middle triplets.** (Fig. 3c) do not have an edge between the resolved nodes. For any middle triplet  $T(r_1, r_2, u)$ ,  $r_1$  and  $r_2$  are not known to share any HLA types since  $(r_1, r_2) \notin E_i$ . Denoting types by  $r_1(\alpha, \beta)$  and  $r_2(\gamma, \delta)$ , all HLA type pairs  $(\rho, \tau)$  where  $\rho \in \{\alpha, \beta\}$  and  $\tau \in \{\gamma, \delta\}$  are assigned as possible HLA types of  $u$ . Each type-pair maintains the consistency of the edges. The triplet in Fig. 4c(I) defines  $(\alpha, \gamma), (\alpha, \delta), (\beta, \gamma)$  and  $(\beta, \delta)$  as possible HLA type-pairs.

If any of  $\alpha = \gamma, \alpha = \delta, \beta = \gamma$  or  $\beta = \delta$  is true, then it is an indication of a false negative. Again, we add the edge  $(r_1, r_2)$  to  $E_i$  and the triplet is treated as a triangle triplet. For example, the triplet in (Fig. 4C(II)) defines  $\alpha$  and  $\beta$  as possible HLA types and triplet in (Fig. 4C(III)) defines possible HLA type  $\alpha$  and possible HLA type-pair  $(\beta, \delta)$ .

Lastly, unresolved nodes may be connected to resolved nodes that are **homozygous** in HLA alleles. If  $(u, r) \in E_i$  where the HLA types of  $r(\alpha, \alpha)$ , the triplet containing  $(u, r) \in E_i$  defines  $\alpha$  as a possible type for  $u$ .

### 3 Algorithms

#### 3.1 Triplet Generation

The edges of  $G_{\text{IBD}}^i$  are represented using the adjacency list representation. More precisely, for efficiency reasons, any given individual stores two adjacency lists, for resolved and unresolved neighbors, respectively.

The algorithm for triplet generation is formally described in Fig. 5. Briefly, triplets are generated by traversing the graph for all paths of length 3 containing only one unresolved individual. Each traversal starts from a resolved individual  $r$ , and progress in two ways based on the status of the adjacent individual,  $a$ :

1. If  $a$  is resolved, traverse through all the unresolved adjacent individuals of  $a$ . This will generate candidate end triplets.
2. If  $a$  is unresolved, traverse through all the resolved adjacent individuals of  $a$  to generate candidate middle triplets and traverse through all resolved adjacent individuals of  $r$  to generate candidate end triplets.

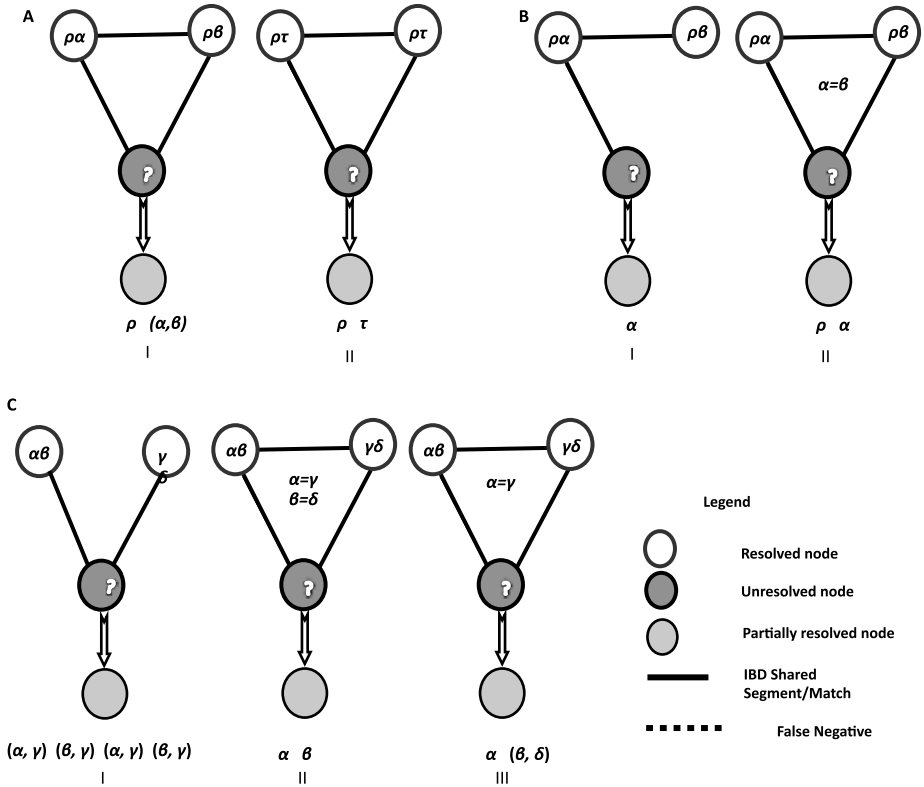
If a trio of individuals generates both end and middle candidate triplets, a *triangle triplet* of the trio is added. Other candidate end or middle triplets are indeed end or middle triplets respectively. Duplicate generated triplets are identified and removed. The algorithm then proceeds to resolve individuals.

#### 3.2 Type Resolution

Define  $S = \{t, e, m, f, h\}$  as the categories of information, representing triangle triplets, end triplets, middle triplets, false negatives and homozygous matches respectively.

Type resolution assigns the most likely HLA types to an unresolved individual. Define  $C_s(\alpha)$  as the number of instances of category  $s$  defining  $\alpha$  as a type for the unresolved node,  $u$  under examination.

The quintuple  $(C_t(\alpha), C_e(\alpha), C_m(\alpha), C_f(\alpha), C_h(\alpha))$  is the sufficient statistic for calculating the likelihood of  $\alpha$  being assigned as follows: Define  $L_t(s)$  to be the likelihood of triplet of category  $s$  being correct and  $L_f(s)$  as likelihood of triplet of category  $s$  being incorrect. Define  $C_s^u$  to be the total number of triplets



**Fig. 4. Possible type from triplets.** (A) Possible type generation for triangle triplets. (I) One shared type: Shared type or the combination of non shared types. (II) Two shared types. (B) Possible type generation for end triplets. (I) One shared type: The non shared type. (II) False negative: Two shared types. (C) Possible type generation for middle triplets. (I) Consistent edges: Combinations of the types of resolved individuals. (II) False negative with two shared types. (III) False negative with one shared type: Shared type and the combination of non shared types

of category  $s$  for unresolved individual  $u$ . The likelihood of  $\alpha$  being the resolution for individual  $u$  is calculated as

$$Likelihood(\alpha|u, Counts) = \prod_{s \in S} L_t(s)^{C_s(\alpha)} \prod_{s \in S} L_f(s)^{C_s^u - C_s(\alpha)} \quad (1)$$

For HLA type-pairs  $(\alpha, \beta)$ , define  $E_s(\alpha, \beta)$  to be the effective count triplets of category  $s$  defining  $(\alpha, \beta)$  as a possible HLA type-pair. The likelihood calculation of HLA type-pair uses the same formula but after calculating the effective counts given by

$$E_s(\alpha, \beta) = C_s(\alpha) + C_s(\beta) - C_s(\alpha, \beta) \quad (2)$$

```

GENERATE-TRIPLET ( $G_{\text{IBD}}^i, R_{i-1}$ ):
 $G_{\text{IBD}}^i$ : IBD-Graph
 $R_{i-1}$ : Set of individuals resolved in iteration (i-1)
define sets  $T_t, T_e, T_m$ : Set of triangle, end
and middle triplets respectively

for  $r$  in  $R_{i-1}$  do
  for  $v$  s.t  $(r, v) \in E^i$ 

      if  $v.RESOLVED = TRUE$ 
      then
           $T'_e = T'_e \cup \{(r, v, u) | (v, u) \in G_{\text{IBD}}^i, u.RESOLVED = FALSE\}$ 
      else
           $T'_m = T'_m \cup \{(r, v, w) | (v, w) \in G_{\text{IBD}}^i, w.RESOLVED = TRUE\}$ 
           $T'_e = T'_e \cup \{(r, v, w) | (v, w) \in G_{\text{IBD}}^i, w.RESOLVED = FALSE\}$ 

 $T_t = T'_e \cap T'_m$ 
 $T_e = T'_e \setminus T_t$ 
 $T_m = T'_m \setminus T_t$ 

```

**Fig. 5.** Algorithm for triplet generation

where  $C_s(\alpha, \beta)$  represents the triplets of category  $s$ , defining both  $\alpha$  and  $\beta$  as possible types.

The score for HLA type-pair  $(\alpha, \beta)$  and individual  $u$  is calculated as

$$Likelihood((\alpha, \beta) | u, \text{Counts}) = \prod_{s \in S} L_t(s)^{E_s(\alpha, \beta)} \prod_{s \in S} L_f(s)^{C_s^u - E_s(\alpha, \beta)} \quad (3)$$

Define  $p^+, q^+, p^-$  and  $q^-$  to be the likelihoods of true positive, false positive, true negative and false negatives edges respectively. The likelihoods for the triplets being correct or incorrect are calculated as in Table [1](#).

**Table 1.** Likelihood terms for correct and incorrect triplets

Category	Likelihood of correct triplet	Likelihood of incorrect triplet
Triangle triplet	$(p^+)^3$	$(q^+)^3$
End triplet, Middle triplet	$(p^+)^2 \times p^-$	$(q^+)^2 \times q^-$
False negative	$(p^+)^2 \times q^-$	$(q^+)^2 \times p^-$
Triangle triplet	$p^+$	$q^+$

We estimate the following rates from GERMLINE  $p^+ : 0.9, p^- : 0.85, q^+ : 0.1, q^- : 0.15$  [\[4\]](#).

The algorithm for type resolution is formally described in Fig. [6](#). The algorithm greedily resolves individuals by the likelihood calculation. Our implementation maintains a hash-map of all possible HLA types and type-pairs for each

individual. The value of the hash-map is the quintuple, with the type or the type-pair being the key.

The most likely HLA type,  $\eta$  is first chosen. If the individual is determined to be homozygous genotypically,  $\eta$  is assigned as the resolution for the individual. If  $\eta$  satisfies all the edges with the resolved individuals, the individual is considered *potentially homozygous* in HLA types. In such cases, the individual is left unresolved and retained for processing in the further iterations.

HLA type-pairs are formed by combining each possible HLA type with  $\eta$ . The two most likely type-pairs are determined and the HLA type-pair with highest likelihood is assigned as resolution, if the difference between their likelihoods is greater than zero.

At the end of each iteration, the adjacency lists of individuals containing newly resolved individuals are updated to move the newly resolved individuals to the head of the list. Thus the entire adjacency matrix needs to be constructed only once at the start of the algorithm. All the steps are repeated until convergence where no more resolutions are possible.

### 3.3 Complexity and Implementation

Triplet generation explores all paths of length three containing only one unresolved node. Thus the time complexity for triplet generation can be estimated as  $O(|R|^2|V \setminus R|)$  where  $R$  is the set of resolved nodes and  $V$  is the set of all nodes in the graph.

Type resolution is linearly dependent on the number of triplets generated since each triplet is examined only once to identify the possible HLA types and type-pairs. Thus  $O(|R|^2|V \setminus R|)$  is the bound on complexity.

The program was implemented in Java 1.5 and testing was done on a Linux node of  $2 \times 2.4$  GHz Xeon CPUs with 2 GB of memory. The average runtimes per individual for cross validation of HLA-A, HLA-B, HLA-C, HLA-DRB1, HLADQA1, HLADQB1 genes were 15, 4, 3, 0.7, 0.7, 1.3 seconds respectively analyzing 5475, 3328, 3387, 2899, 2545, 2426 IBD shared segments. The software has been made available for download at [http://www1.cs.columbia.edu/~itsik/hla\\_ibd/index.html](http://www1.cs.columbia.edu/~itsik/hla_ibd/index.html).

## 4 Data

The data used for analysis has been described in [3]. Briefly, The data includes 90 individuals (30 parent-offspring trios) of the Yoruba people from Ibadan, Nigeria (YRI); 182 Utah residents (29 extended families of European ancestry, from the Centre d'Etude du Polymorphisme 6 is available. HLA typing was carried out for class I (HLA-A, HLA-B, HLA-C) and class II (HLA-DRB1, HLA-DQA1, HLA-DQB1) genes using the PCR-SSOP protocols. The CEU and YRI populations were used for analysis of the model and all the data was assumed to be unphased for *Centre d'Etude du Polymorphisme Humain (CEPH)* collection (CEU); 45 unrelated Han Chinese from Beijing, China (CHB); and 44 unrelated

RESOLUTION ( $U$ ) $U$ : Unresolved individuals

```

for  $u$  in  $U$  do
  determine  $\eta$ : highest scoring possible type
  if  $u$  is homozygous
  then
     $u.TYPES := (\eta, \eta)$ 
     $u.RESOLVED := TRUE$ 
  else
    if  $\eta$  satisfies all the matches with resolved  $r : (u, r) \in G_{IBD}^i$ 
    then
       $u$  cannot be completely resolved in the current iteration
       $u.RESOLVED := FALSE$ 
      this is an indication of possible homozygous HLA types
    else
      for each possible type  $\alpha$  do
        if  $\tau \neq \eta$ 
        then
          define type-pair  $(\alpha, \eta)$ 
          False negative detection based on  $\eta$  (Fig.7)
          Calculate LIKELIHOOD( $(\alpha, \eta)$ )

      find  $(\alpha, \tau)$  and  $(\beta, \tau)$ , the two most likely type-pairs
      if LIKELIHOOD( $(\alpha, \eta)$ )  $\neq$  LIKELIHOOD( $(\beta, \eta)$ )
      then
         $u.TYPES := (\alpha, \tau)$ 
         $u.RESOLVED := TRUE$ 
      else
         $u.RESOLVED := FALSE$ 

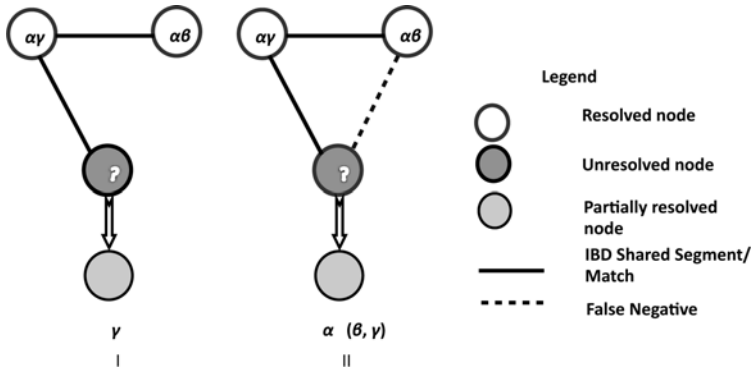
```

**Fig. 6.** Algorithm for type resolution

Japanese from Tokyo, Japan(JPT), 6338 variants located in a 7.5Mb region on chromosome 6 is available. HLA typing was carried out for class I (HLA-A, HLA-B, HLA-C) and class II (HLA-DRB1, HLA-DQA1, HLA-DQB1) genes using the PCR-SSOP protocols. The CEU and YRI populations were used for analysis of the model and all the data was assumed to be unphased for analysis.

## 5 Results

We establish the accuracy of our method on the CEU HapMap data. Intuitively, the effectiveness of the model is dependent on the presence of IBD among the individuals under consideration. Leave one out cross-validation offers a good approach to test the model since it utilizes all the available IBD shared segment information for the individual being tested. An individual being tested is instantiated as unresolved and all the other individuals form the resolved individuals or training set for the model. The individual can either be inferred as resolved,



**Fig. 7. Illustration of end triplets detecting false negatives.**(I) The generated end triplet. (II) The same end triplet becomes a false negative if  $\beta$  is chosen as  $\gamma$ .

if types on both chromosomes are inferred; as ambiguous, if two types cannot be inferred or more than two types are equally likely; as potentially homozygous, if only a single type is present and inferred; or as unresolved, if no inference can be made. The model is not dependent on learning any parameters from the training data and therefore leave one out cross-validation does not bias the results.

Each locus is analyzed separately and the accuracy and coverage are defined with respect to the number of chromosomes analyzed. Formally, let  $u(\rho, \tau)$  be the individual being tested. If  $u$  is inferred as resolved with HLA types  $(\alpha, \beta)$ , both the chromosomes are accounted as called and both are correct if  $(\alpha, \beta) = (\rho, \tau)$ . Only one of the chromosomes is considered correct if  $\alpha \in (\rho, \tau)$  and  $\beta \notin (\rho, \tau)$  or vice-versa. If  $u$  is inferred as ambiguous with HLA type  $\alpha$ , one chromosome is called and is correct if  $\alpha = \rho$  or  $\alpha = \tau$ . If  $u$  is inferred as potentially homozygous with type  $\alpha$ , both chromosomes are called; both are correct if  $\alpha = \rho = \tau$ , one is correct if  $\rho \neq \tau$  and,  $\alpha = \rho$  or  $\alpha = \tau$ .

The coverage and accuracy are measures as follows

$$\text{Coverage} = \frac{\text{TotalCalled}}{\text{TotalAnalyzed}} \quad (4)$$

$$\text{Accuracy} = \frac{\text{TotalCorrect}}{\text{TotalCalled}} \quad (5)$$

Table 2 lists the results of leave one out cross-validation tests on the CEU HapMap population. The analysis examined shared segments which span 100kb upstream and downstream of the gene under consideration. Results are shown for both four-digit and two-digit resolutions. HLA types occurring only once in the population and types which are not resolved to the required extent are excluded from analysis. The model predicts results with high accuracy in the HLA-A, HLA-B, HLA-C, HLA-DQA1 and HLA-DQB1 alleles at four-digit resolution. The accuracy for class I genes remains the approximately the same at both the resolutions, but using two-digit resolution leads to higher accuracy in

**Table 2.** Results of leave one out cross-validation for CEU population by considering IBD shared segments which span 100kb upstream and downstream of the gene

Gene	Four-digit		Two-digit	
	Analyzed	Accuracy	Analyzed	Accuracy
HLAA	314	96.5	322	96.4
HLAB	281	94.3	311	93.4
HLAC	328	94.2	316	94.9
HLADRB1	308	77.6	294	91.3
HLADQA1	350	92.6	330	94.3
HLADQB1	350	90.3	323	92.3

the class II genes. This can be attributed to a reduction in the false positive matches at lower resolution.

The main sources of error are false positives and non-availability/non-detection of IBD between individuals. The distribution of the false positives at four-digit resolution for the different genes is illustrated in Fig. 8. For each individual, the false positive percentage in a region is the percentage of matches of the resolved adjacent nodes which are false positive. The HLA-DRB1 region has a higher number of individuals with large false positive percentages which is reflected in the low accuracy prediction. Fitting the parameters of IBD detection, especially the specific span used will improve results, but we chose to present benchmarks with vanilla parameters across all genes.

We compare our results to the results from phasing the data. Phased version of the CEU data based on trios is available from [3]. We used GERMLINE [4] to obtain pairwise IBD segments between the haplotypes. Leave one out crossvalidation is again used for testing. If  $M$  is the set of matches determined by GERMLINE, the likelihood of allele  $\alpha$  being the resolution for chromosome  $c$  is calculated as below

$$Likelihood(\alpha|c, M) = \frac{H(\alpha)}{\sum_{\beta \in A} H(\beta)} \quad (6)$$

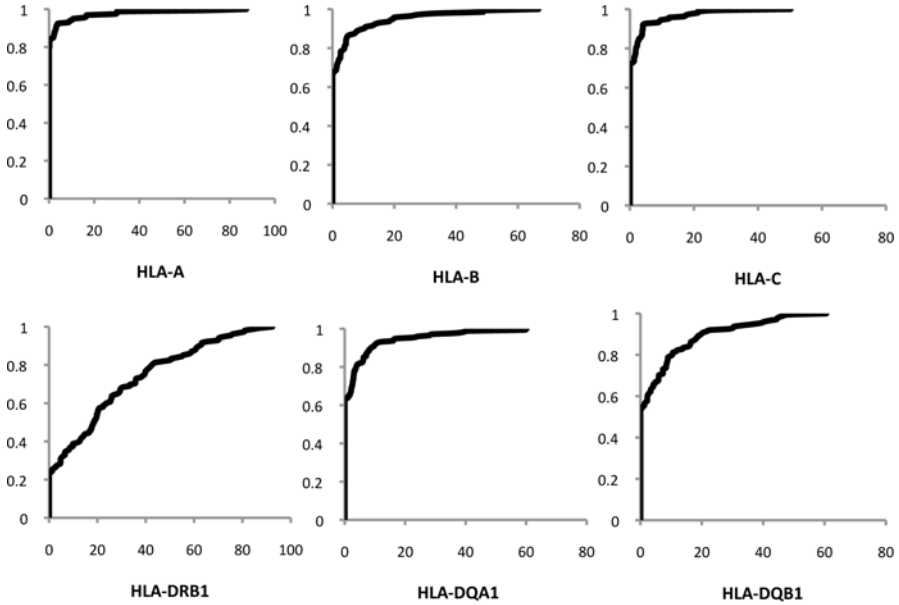
where  $H(\alpha) = \sum_{(y,c) \in M} \delta(\beta_y, \alpha)$  where  $\beta_y$  is the HLA type of chromosome  $y$  and  $\delta$  function given by

$$\delta(\alpha, \beta) = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases} \quad (7)$$

The HLA type with highest likelihood as assigned as resolution. If two or more HLA types are tied for the highest likelihood, the chromosomes is left unresolved and considered ambiguous.

We also used fastPHASE [14] to perform phasing without using the trio information and again use GERMLINE to determine set of matches using haplotypic extensions of matches rather than genotypic extensions. The triplet based algorithm is used to determine the resolutions in a leave one out crossvalidation setting.





**Fig. 8. False positive rate distribution.** X-axis represents the false positive percentage and Y-axis represents the fraction of individuals. Each point on the graph represents the fraction of individuals with false positive percentage less than the corresponding reference value.

The performance is assessed by means of the sensitivity and specificity differences. If  $A$  is the set of alleles under examination and  $P_{\alpha}^{+}, P_{\alpha}^{-}, N_{\alpha}^{+}$  and  $N_{\alpha}^{-}$  are the positive, false positive, true negative and false negatives for allele  $\alpha$  respectively, sensitivity and specificity are calculated as below.

$$Sensitivity = \frac{\sum_{\alpha \in A} P_{\alpha}^{+}}{\sum_{\alpha \in A} (P_{\alpha}^{+} + N_{\alpha}^{-})} \tag{8}$$

$$Specificity = \frac{\sum_{\alpha \in A} N_{\alpha}^{+}}{\sum_{\alpha \in A} (N_{\alpha}^{+} + P_{\alpha}^{-})} \tag{9}$$

The comparison plot is shown in Fig. 9. All the methods show very high specificity ( $> 0.95$  in all the genes). The performance of our method compares well with trio based phased data results in the class I genes whereas having phased data has significant benefits of the class II genes. This could be possibly because of a reduction in the false positive rates in IBD matches when using trio based phasing. This demonstrates that our method can be applied to unphased data with accuracy comparable to phased data when the false positive rates in the IBD segment determination are low. Phasing computationally without using the trio information performs worse in both class I and class II genes demonstrating the effectiveness of using genotypic extension when trio based phased data is not available.

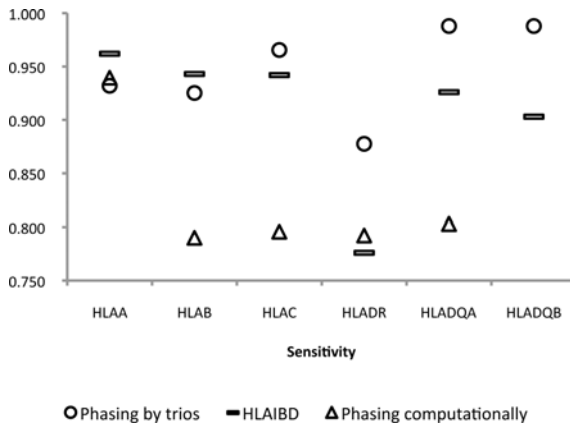


Fig. 9. Sensitivity of the different approaches

## 6 Discussion

We have developed a method for inferring HLA types using genotypic data by examining IBD shared segments with individuals of known HLA types. HLA types are predicted with high accuracy by using the CEU HapMap data. HLA genes play a critical role in adaptive immune response and autoimmune diseases. Our model can be used as a starting point in the analysis of similar diseases. The further applicability of SNP based methods of HLA type inference has been described elsewhere [7].

Although advances have been made in phasing, inferring haplotype structure in small cohorts or unrelated individuals remains a challenge [9]. Our method analyses genotypic data without consulting the haplotype phases. This broadens the applicability of the method to data where the phase is unknown.

## References

1. Breese, E., Braegger, C.P., Corrigan, C.J., Walker-Smith, J.A., MacDonald, T.T.: Interleukin-2- and interferon-gamma-secreting T cells in normal and diseased human intestinal mucosa. *Immunology* 78(1), 127–131 (1993)
2. Brimnes, J., Allez, M., Dotan, I., Shao, L., Nakazawa, A., Mayer, L.: Defects in CD8+ regulatory T cells in the lamina propria of patients with inflammatory bowel disease. *J. Immunol.* 174, 5814–5822 (2005)
3. de Bakker, P.I.W., et al.: A high-resolution HLA and SNP haplotype map for disease association studies in the extended human MHC. *Nature Genetics* 38(10), 1166–1172 (2006)
4. Gusev, A., et al.: Whole Population, Genome-Wide Mapping of Hidden Relatedness. *Genome Research* 19(2), 318–326 (2008)
5. Horton, R., et al.: Gene map of the extended human MHC. *Nature Reviews Genetics* 5, 889–899 (2004)

6. Kong, A., et al.: Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature Genetics* 40(9), 1068–1075 (2008)
7. Leslie, S., Donnelly, P., McVean, G.: A new statistical method for predicting classical HLA alleles from SNP data. *Am. J. Hum. Genet.* 82(1), 48–56 (2008)
8. Lincoln, M., et al.: A predominant role for the HLA class II region in the association of the MHC region with multiple sclerosis. *Nature Genetics* 37(10), 1108–1112 (2005)
9. Marchini, J., et al.: A Comparison of Phasing Algorithms for Trios and Unrelated Individuals. *Am. J. Hum. Genet.* 78(3), 437–450 (2006)
10. Marsh, S.G., et al.: Nomenclature for Factors of the HLA System, 2004. *Tissue Antigens* 65, 301–369 (2005); *Human Immunology* 66, 571–636 (2005); *International Journal of Immunogenetics* 32, 107–159 (2005)
11. Miretti, M.M., et al.: A high-resolution linkage-disequilibrium map of the human major histocompatibility complex and first generation of tag single-nucleotide polymorphisms. *Am. J. Hum. Genet.* 76(4), 634–646 (2006)
12. Purcell, S., et al.: PLINK: a toolset for whole-genome association and population-based linkage analysis. *Am. J. Hum. Genet.* 81(3), 559–575 (2007)
13. Robinson, J., Waller, M.W., Parham, P., Bodmer, J.G., Marsh, S.G.E.: IMGT/HLA Database - sequence database for the Human Major Histocompatibility Complex. *Tissue Antigens* 55, 280–287 (2000)
14. Scheet, P., Stephens, M.: A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am. J. Hum. Genet.* 78(4), 629–644 (2006)

# Algorithms for Detecting Significantly Mutated Pathways in Cancer

Fabio Vandin<sup>1,\*</sup>, Eli Upfal<sup>2</sup>, and Benjamin J. Raphael<sup>2,3,\*\*</sup>

<sup>1</sup> Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy  
vandinfa@dei.unipd.it

<sup>2</sup> Department of Computer Science, Brown University, Providence, RI  
{eli,braphael}@cs.brown.edu

<sup>3</sup> Center for Computational Molecular Biology, Brown University, Providence, RI

**Abstract.** Recent genome sequencing studies have shown that the somatic mutations that drive cancer development are distributed across a large number of genes. This mutational heterogeneity complicates efforts to distinguish functional mutations from sporadic, passenger mutations. Since cancer mutations are hypothesized to target a relatively small number of cellular signaling and regulatory pathways, a common approach is to assess whether known pathways are enriched for mutated genes. However, restricting attention to known pathways will not reveal novel cancer genes or pathways. An alternative strategy is to examine mutated genes in the context of genome-scale interaction networks that include both well characterized pathways and additional gene interactions measured through various approaches. We introduce a computational framework for *de novo* identification of subnetworks in a large gene interaction network that are mutated in a significant number of patients. This framework includes two major features. First, we introduce a diffusion process on the interaction network to define a local neighborhood of “influence” for each mutated gene in the network. Second, we derive a two-stage multiple hypothesis test to bound the false discovery rate (FDR) associated with the identified subnetworks. We test these algorithms on a large human protein-protein interaction network using mutation data from two recent studies: glioblastoma samples from The Cancer Genome Atlas and lung adenocarcinoma samples from the Tumor Sequencing Project. We successfully recover pathways that are known to be important in these cancers, such as the p53 pathway. We also identify additional pathways, such as the Notch signaling pathway, that have been implicated in other cancers but not previously reported as mutated in these samples. Our approach is the first, to our knowledge, to demonstrate a computationally efficient strategy for *de novo* identification of *statistically significant* mutated subnetworks. We anticipate that our approach will find increasing use as cancer genome studies increase in size and scope.

---

\* Supported in part by the “Ing. Aldo Gini” Foundation, Padova, Italy. This work was done while the author was visiting the Department of Computer Science of Brown University.

\*\* BJR is supported by a Career Award at the Scientific Interface from the Burroughs Wellcome Fund.

## 1 Introduction

Cancer is a disease that is largely driven by somatic mutations that accumulate during the lifetime of an individual. Decades of experimental work have identified numerous cancer-promoting oncogenes and tumor suppressor genes that are mutated in many types of cancer. Recent cancer genome sequencing studies have dramatically expanded our knowledge about somatic mutations in cancer. For example, large projects like The Cancer Genome Atlas (TCGA) [31], the Tumor Sequencing Project (TSP) [8], and the Cancer Genome Anatomy Project [11] have sequenced hundreds of protein coding genes in hundreds of patients with a variety of cancers. Other efforts have taken a global survey of approximately 20,000 genes in a 1-2 dozen patients [40,18,32]. These studies have shown that: tumors harbor on average approximately 80 somatic mutations; two tumors rarely have the same complement of mutations; and thousands of genes are mutated in cancer [40]. This mutational heterogeneity complicates efforts to distinguish functional mutations from sporadic, passenger mutations. While a few cancer genes are mutated at high frequency (e.g. well known cancer genes like TP53 or KRAS), most cancer genes are mutated at much lower frequencies. Thus, the observed frequency of mutation is an inadequate measure of the importance of a gene, particularly with the relatively modest number of samples that are tested in current cancer studies.

It is widely accepted that cancer is a disease of pathways and it is hypothesized that somatic mutations target genes in a relatively small number of regulatory and signaling networks [12,39]. Thus, the observed mutational heterogeneity is explained by the fact that there are myriad combinations of alterations that cancer cells can employ to perturb the behavior of these key pathways. The unifying themes of cancer are thus not solely revealed by the individual mutated genes, but by the interactions between these genes. Standard practice in cancer sequencing studies is to assess whether genes that are mutated at sufficiently high frequency significantly overlap known cancer pathways [31,8,36,40,32,25].

Finding significant overlap between mutated genes and genes that are members of known pathways is an important validation of existing knowledge. However, restricting attention to these known pathways does not allow one to detect novel group of genes that are members of less characterized pathways. Moreover, it is well known that there is crosstalk between different pathways [39] and dividing genes into discrete pathway groupings limits the ability to detect whether this crosstalk is itself a target of mutations. An additional source of information about gene and protein interactions is large-scale interaction networks, such as the Human Protein Reference Database (HPRD) [22], STRING [17], and others [2,34]. These resources incorporate both well-annotated pathways and interactions derived from high-throughput experiments, automated literature mining, cross-species comparisons, and other computational predictions. Many researchers have used these interaction networks to analyze gene expression data. Ideker et al. [16] introduced a method to discover subnetworks of differentially expressed genes, and this idea was later extended in different directions by others [30,26,38,21,28,13,5].

We propose to identify “significantly mutated subnetworks” – that is connected subnetworks whose genes have more mutations than expected by chance – *de novo* in a large gene interaction network. This problem differs from the gene expression problem in that a relatively small number of genes might be measured, a small subset of genes in a pathway may be mutated, and that a single mutated gene may be sufficient to perturb a pathway. The naïve approach to *de novo* identification of mutated subnetworks is to examine mutations on all subnetworks, or all subnetworks of a fixed size. This approach is problematic. First, the enumeration of all such subnetworks is prohibitive for subnetworks of a reasonable size. Second, the extremely large number of hypotheses that are tested makes it difficult to achieve statistical significance. Finally, biological interaction networks typically have small diameter due to the presence of “hub” genes of high degree. There are reports that cancer-associated genes have more interaction partners than non-cancer genes [25,19], and indeed highly mutated cancer genes like TP53 have high degree in most interaction networks (e.g. the degree of TP53 in HPRD is 238). Such correlations might lead to a large number of “uninteresting” subnetworks being deemed significant.

We propose a rigorous framework for *de novo* identification of significantly mutated subnetworks and employ two strategies to overcome the difficulties described above. First, we formulate an *influence* measure between pairs of genes in the network using a diffusion process defined on the graph. This quantity considers a gene to influence another gene if they are both close in distance on the graph *and* there are relatively few paths between them in the interaction network. We use this measure to build a smaller *influence graph* that includes only the mutated genes but encodes the neighborhood information from the larger network. We then identify significant subnetworks using two techniques. The first one requires to solve an NP-hard problem, while in the second one, in which the influence between pairs of genes is enhanced by the number of mutations observed on these genes, the computational problem is reduced to just finding connected components in the graph. Finally, we derive a *two-stage multiple hypothesis test* that mitigates the testing of a large number of hypotheses by focusing on the number of discovered subnetworks of a given size rather than on individual subnetworks. We also show how to estimate the false discovery rate (FDR) associated with this test.

We tested our approach on the HPRD human interaction network using somatic mutation data from two recently published studies: (i) 601 genes in 91 glioblastoma multiforme patients from The Cancer Genome Atlas (TCGA) project; (ii) 623 genes in 188 lung adenocarcinoma patients sequenced during the Tumor Sequencing Project (TSP). In both datasets, we identify statistically significant mutated subnetworks that are enriched for genes on pathways known to be important in these cancers. Our approach is the first, to our knowledge, to demonstrate a computationally efficient strategy for *de novo* identification of *statistically significant* mutated subnetworks. We anticipate that our approach will find increasing use as cancer genome studies increase in size and scope.

## 2 Methods

In this section we introduce our approach for the identification of significantly mutated pathways in cancer. Due to space constraints, the proofs of theorems are omitted. Supplementary material including details of proofs is available at <http://www.cs.brown.edu/people/braphael/supplements/>.

### 2.1 Mathematical Model

We model the interaction network by a graph  $G = (V, E)$ , where the vertices in  $V$  represent individual proteins (and their associated genes), and the edges in  $E$  represent (pairwise) protein-protein or protein-DNA interactions. Let  $\mathcal{T} \subseteq V$  be the subset of genes that have been tested, or assayed, for mutations in a set  $\mathcal{S}$  of samples (patients). The size of  $\mathcal{T}$  will vary by study; e.g. some recent works resequenced hundreds of genes [31,8] while others examine nearly all known protein-coding genes in the human genome [40,18,32]. We assume that each gene  $g$  is assigned one of two labels, *mutated* or *normal*, in each sample. Let  $M_i$  denote the subset of genes in  $\mathcal{T}$  that are mutated in the  $i$ th sample, for  $i = 1, \dots, |\mathcal{S}|$ . Let  $\mathcal{S}_j$  be the samples in which gene  $g_j \in \mathcal{T}$  is mutated, for  $j = 1, \dots, |\mathcal{T}|$ , let  $m = \sum_i |M_i|$  be the total number of occurrences of altered genes observed in all samples.

We define a *pathway* or *subnetwork* to be a connected subgraph of  $G$ . Note that this definition matches the common biological usage of the term where pathways may have arbitrary topology in the graph, and are not restricted to be linear chains of vertices. We generally do not know whether more than one gene must be mutated to perturb a pathway in a sample, and thus will assume that a pathway is mutated in a sample if *any* of the genes in the pathway are mutated. For a subset  $T \subseteq \mathcal{T}$ , let  $S(T)$  denote the set of samples in which *at least one* gene in  $T$  is mutated.

### 2.2 Influence Graph

Our goal is to identify subnetworks that are significant with respect to the set of mutated genes in the samples. The significance of a subnetwork is derived from: (i) the number of samples that have mutations in the genes of the subnetwork, and (ii) the interactions between genes in the subnetwork in the context of the whole network topology. For example, consider two possible scenarios of mutated nodes (Figure 1). In the first scenario, the two mutated nodes are part of a linear chain in the interaction network. In the second scenario, the two mutated nodes are connected through a high-degree node. In the first scenario, there is a single path joining the two mutated nodes and thus we are more surprised by this local clustering of mutations than in the second scenario, where the two nodes are connected by a node that is present in a large number of possible paths.

Hubs present an extreme case of this phenomenon and result in many “uninteresting” subnetworks being deemed significant. Since many highly mutated cancer genes, like TP53, also have high degree in interaction networks it is not advisable

to ignore these genes in the analysis of cancer mutation data. These examples show that significance of a subnetwork is derived from both: 1. the number of samples that have mutations in the genes of the subnetwork, and 2. the interactions between genes in the subnetwork in the context of the whole network. A straightforward graph distance like the shortest path between nodes is not sufficient to overcome the problems highlighted above. Moreover, other graph mining approaches like dense subgraph identification [10] are also not appropriate, since not all subnetworks of interest (e.g. the chain in Figure 1) are dense in edges.

We use a diffusion process on the interaction network to define a rigorous measure of *influence* between all pairs of nodes. To measure the influence of node  $s$  on all the other nodes in the graph, consider the following process, described by [33]. Fluid is pumped into the source node  $s$  at a constant rate, and fluid diffuses through the graph along the edges. Fluid is lost from each node at a constant first-order rate  $\gamma$ . Let  $f_v^s(t)$  denote the amount of fluid at node  $v$  at time  $t$ , and let  $\mathbf{f}^s(t) = [f_1^s(t), \dots, f_n^s(t)]^T$  be the column vector of fluid at all nodes. Let  $L$  be the Laplacian matrix of the graph<sup>1</sup>, and let  $L_\gamma = L + \gamma I$ . Then the dynamics of this continuous-time process are governed by the vector equation  $\frac{d\mathbf{f}^s(t)}{dt} = -L_\gamma \mathbf{f}^s(t) + \mathbf{b}^s u(t)$ , where  $\mathbf{b}^s$  is the elementary unit vector with 1 at the  $s^{\text{th}}$  place and 0 otherwise, and  $u(t)$  is the unit step function. As  $t \rightarrow \infty$ , the system reaches the steady state. The equilibrium distribution of fluid density on the graph is  $\mathbf{f}^s = L_\gamma^{-1} \mathbf{b}^s$  (see [33]). Note that this diffusion process is related to the diffusion kernel [24], or heat kernel [6], which models the diffusion of heat on a graph, and these diffusion processes are also related to certain random walks on graphs [9,27]. Diffusion processes and their related flow problems have been used in protein function prediction on interaction networks [37,29] and to define associations between gene expression and phenotype [28].

We interpret  $f_i^s$  as the influence of gene  $g_s$  on gene  $g_i$ . Computing the diffusion process for all tested genes gives us, for each pair of genes  $g_j, g_k \in \mathcal{T}$ , the influence  $i(g_j, g_k)$  that gene  $g_j$  has on gene  $g_k$ . Note that in general the influence is not symmetric; i.e.  $i(g_j, g_k) \neq i(g_k, g_j)$ . We define an *influence graph*  $G_I = (\mathcal{T}, E_I)$  with the set of nodes corresponding to the set of tested genes, the weight of an edge  $(g_j, g_k)$  is given by  $w(g_j, g_k) = \min[i(g_k, g_j), i(g_j, g_k)]$ . If  $n$  is the number of nodes in the interaction network, then the cost of computing  $G_I$  is dominated by the complexity of inverting an  $n \times n$  matrix.

### 2.3 Discovering Significant Subnetworks: Combinatorial Model

Given an influence measure between genes, the obvious first approach for discovering significant subnetworks is to identify sets of nodes in the influence graph

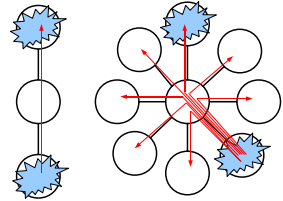


Fig. 1. Mutation on chain vs. star graph

<sup>1</sup>  $L = -A + D$ , where  $A$  is the adjacency matrix of the graph and  $D$  is a diagonal matrix with  $D_{i,i} = \text{degree}(v_i)$ .



$G_I$  that are (1) connected through edges with high influence measure; and (2) correspond to mutated genes in a significant number of samples. We fix a threshold  $\delta$  and compute a *reduced influence graph*  $G_I(\delta)$  of  $G_I$  by removing all edges with  $w(g_i, g_j) < \delta$ , and all nodes corresponding to genes with no mutations in the sample data. The computational problem is reduced to identifying connected subgraphs of  $G_I(\delta)$  such that the corresponding set of genes is altered in a significant number of patients.

The size of the connected subgraphs we discover is controlled by the threshold  $\delta$ . We choose sufficiently small  $\delta$  such that in the null hypothesis, in which the mutations are randomly placed in nodes corresponding to tested genes, it is unlikely that our procedure finds connected subgraphs with similar properties. Note that value of  $\delta$  depends only on the null hypothesis and not on the observed sample data (see Section 2.5 for details of the statistical analysis). Finding the connected subgraph of  $k$  genes that is mutated in the largest number of samples requires to solve the following problem, that we define as *connected maximum coverage* problem.

**Computational Problem.** Given a graph  $G$  defined on a set of  $m$  vertices  $V$ , a set of elements  $I$ , a family of subsets  $\mathcal{P} = \{P_1, \dots, P_m\}$ , with  $P_i \in 2^I$  associated to  $v_i \in V$ , and a value  $k$ , find the connected subgraph  $\mathcal{C}^* = \{v_{i_1}, \dots, v_{i_k}\}$  with  $k$  nodes in  $G$  that maximize  $|\cup_{j=1}^k P_{i_j}|$ . In our case we have  $G = G_I(\delta)$ ,  $V$  is the subset of genes in  $\mathcal{T}$  mutated in at least one sample, and for each  $g_i \in V$  the associated set is  $\mathcal{S}_i$ . The connected maximum coverage problem is related to the maximum coverage problem (see e.g. [14] for a survey) where given a set  $I$  of elements, a family of subsets  $F \subset 2^I$ , and a value  $k$ , one needs to find a collection of  $k$  sets in  $F$  that covers the maximum number of elements in  $I$ . This problem is NP-hard as set cover is reducible to it.

If the graph  $G$  is a complete graph, the connected maximum coverage problem is the same as the maximum coverage problem. Thus the connected maximum coverage problem is NP-hard for a general graph. Moreover we prove that the problem is still hard even on simple graphs such as the star graph ([35] gives a similar result for the connected set cover problem).

**Theorem 1.** *The connected maximum coverage problem on star graphs is NP-hard.*

Since the connected maximum coverage problem is NP-hard even for simple graphs we turn to approximate solutions. It is not hard to construct a polynomial time  $1 - \frac{1}{e}$  approximation algorithm for spider graphs (analogous to the result in [35] for the connected set cover problem). Since it cannot be applied to the network here, we construct an alternative polynomial time algorithm that gives  $O(1/r)$  approximation when the radius of the optimal solution  $\mathcal{C}^*$  is  $r$ .

Our algorithm obtains a solution  $\mathcal{C}_v$  (thus, a connected subgraph) starting from each node  $v \in V$ , and then returns the best solution found. To obtain  $\mathcal{C}_v$ , our algorithm executes an *exploration phase*, i.e. for each node  $u \in G$  it finds a shortest path  $p_v(u)$  from  $v$  to  $u$ . Let  $\ell_v(u)$  be the set of nodes in  $p_v(u)$ , and  $P_v(u)$  the elements of  $I$  that they cover. After this *exploration phase*, the

algorithm builds a connected subgraph  $\mathcal{C}_v$  starting from  $v$ . At the beginning we have  $\mathcal{C}_v = \{v\}$ .  $P_{\mathcal{C}_v}$  is the set of elements covered by the current connected subgraph  $\mathcal{C}_v$ . Then, while  $|\mathcal{C}_v| < k$ , the algorithm chooses the node  $u \notin \mathcal{C}_v$  such that:  $u = \arg \max_{u \in V} \left\{ \frac{|P_v(u) \setminus P_{\mathcal{C}_v}|}{|\ell_v(u) \setminus \mathcal{C}_v|} \right\}$  and  $|\ell_v(u) \cup \mathcal{C}_v| \leq K$ ; the new solution is then  $\ell_v(u) \cup \mathcal{C}_v$ . The main computational cost of our algorithm is due to the exploration phase, that can be performed in polynomial time. We have the following:

**Theorem 2.** *The algorithm above gives a  $\frac{1}{c^r}$ -approximation for the connected maximum coverage problem on  $G$ , where  $c = \frac{2e-1}{e-1}$  and  $r$  is the radius of optimal solution in  $G$ .*

For our experiments we implemented a variation of this algorithm, that for each pair of nodes  $(u, v)$  considers all the shortest paths between  $u$  and  $v$ , and then keeps the one that maximizes  $\frac{|P_v(u)|}{|\ell_v(u)|}$  to build the solution  $\mathcal{C}_v$ . With this modification the algorithm is not guaranteed to run in polynomial time in the worst-case, but ran efficiently for all our experiments.

## 2.4 Discovering Significant Subnetworks: The Enhanced Influence Model

We developed an alternative, computationally efficient, approach for identifying subnetworks that are significant with respect to the gene mutation data. The *Enhanced Influence Model* is based on the idea of enhancing the influence measure between genes by the number of mutations observed in each of these genes, and then decomposing an associated *enhanced influence graph* into connected components.

We define the *enhanced influence graph*  $H$ . It has a node for each gene  $g_j$  with at least one mutation in the data. The weight of edge  $(g_j, g_k)$  in  $H$  is given by  $h(g_j, g_k) = \min \{i(g_j, g_k), i(g_k, g_j)\} \times \max \{|\mathcal{S}_j|, |\mathcal{S}_k|\}$ . Thus, the strength of connection between two nodes in the enhanced influence graph is a function of both the interaction between the nodes in the interaction network and the number of mutations observed in their corresponding genes. Next we remove all edges with weight smaller than a threshold  $\delta$  to obtain a graph  $H(\delta)$ . We return the connected components in  $H(\delta)$  as the significant subnetworks with respect to the mutation data and the threshold  $\delta$ . The computational cost is the complexity of computing all connected components in a graph with  $|S|$  nodes (number of mutated genes), which is linear in the size of the graph. The significance of the discovered subnetworks depends on the choice of  $\delta$ . We choose sufficiently small  $\delta$  such that in the null hypothesis, in which the mutations are randomly placed in nodes corresponding to tested genes, it is unlikely that our procedure finds connected components of similar size (see Section 2.5 for details).

## 2.5 Statistical Analysis

We assess the statistical significance of our discoveries with respect to null hypothesis distributions in which the mutated genes are randomly allocated in the

network, i.e. when the occurrence of mutations are independent of the network topology. We consider two null hypothesis distributions: in  $H_0^{\text{sample}}$  a total of  $m = \sum_i |M_i|$  mutations are placed uniformly at random in the nodes corresponding to the  $|\mathcal{T}|$  tested genes. While easier to analyze, this model does not account for the fact that in the observed data a large number of mutations are concentrated in a few genes (e.g. TP53). Thus, we also use a second null hypothesis distribution,  $H_0^{\text{gene}}$ , generated by permuting the identities of the tested genes in the network. That is we select a random permutation  $\sigma$  of the set  $\{1, \dots, |\mathcal{T}|\}$ , and we assign gene  $g_j$ , that was mutated in the set of samples  $\mathcal{S}_j \subseteq \mathcal{S}$ , to the location of gene  $g_{\sigma(j)}$  in the original network.

**A Two Stage Multi-Hypothesis Test.** A major difficulty in assessing the statistical significance of the discovered subnetworks is that we test simultaneously for a large number of hypotheses; each connected subnetwork in the interaction graph with at least one tested gene is a possible significant subnetwork and thus an hypothesis. The strict measure of significance level in multi-hypothesis testing is the *Family Wise Error Rate (FWER)*, the probability of incurring at least one Type I error in any of the individual tests. An alternative, less conservative approach to control errors in multiple tests is the *False Discovery Rate (FDR)* [3]. Let  $V$  be the number of Type I errors in the individual tests, and let  $R$  be the total number of null hypotheses rejected by the multiple test. We define  $FDR = E[V/R]$  to be the expected ratio of erroneous rejections among all rejections (with  $V/R = 0$  when  $R = 0$ ). Let  $h$  be the total number of hypothesis tested. Applying either measure to our problem, a discovery would be flagged as statistically significant only if its  $p$ -value is  $O(1/h)$ , which is impractical in the size of our problem. Instead, building on an idea presented in [23], we develop a two stage test for our problem that allows us to flag a number of subnetworks in our data as statistically significant with small false discovery rate (FDR) values.

We demonstrate our method through the analysis of the Enhanced Influence model. A similar technique was applied to the Combinatorial model. Let  $C_1, \dots, C_\ell$  be the set of connected components found in the enhanced influence graph  $H(\delta)$ . Testing for the significance of these discoveries is equivalent to simultaneously testing for  $2^{|\mathcal{T}|}$  hypothesis. To reduce the number of hypothesis we focus on an alternative statistic: the *number* of discoveries of a given size. Let  $\tilde{r}_s$  be the number of connected components of size  $\geq s$  found in the graph  $H(\delta)$ , and let  $r_s$  be the corresponding random variable in the null hypothesis ( $H_0^{\text{sample}}$  or  $H_0^{\text{gene}}$ ). We are testing now for just  $\mathcal{K} = |\mathcal{T}|$  simple hypotheses, for  $s = 1, \dots, \mathcal{K}$ :  $E_s \equiv \tilde{r}_s$  conforms with the distribution of  $r_s$ ". Testing each hypothesis with confidence level  $\alpha/\mathcal{K}$ , the first stage of our test identifies the smallest size  $s$  such that with confidence level  $\alpha$  we can reject the null hypothesis that  $\tilde{r}_s$  conforms with the distribution of  $r_s$ .

The fact that the number of connected components of size at least  $s$  is statistically significant does not imply necessarily that each of the connected components is significant. We now add a second condition to the test that guarantees an upper bound on the False Discovery Rate (FDR):

**Theorem 3.** Fix  $\beta_1, \beta_2, \dots, \beta_{\mathcal{K}}$  such that  $\sum_{i=1}^{\mathcal{K}} \beta_i = \beta$ . Let  $s^*$  be the first  $s$  such that  $\tilde{r}_s \geq \frac{\mathbf{E}[r_s]}{\beta_s}$ . If we return as significant all connected components of size  $\geq s^*$ , then the FDR of the test is bounded by  $\beta$ .

In our tests we have used  $\beta_i = \frac{\beta}{2^i}$  for the  $i^{\text{th}}$  largest  $s$  tested (with  $\beta_s = \beta - \sum_i \beta_i$  for the smallest  $s$ ), since we are more interested in finding large connected components.

**Estimating the Distribution of the Null Hypothesis.** The null hypothesis distributions can be estimated by either a Monte-Carlo simulation (“permutation test”) or through analytical bounds.

Using Monte-Carlo simulation, two features of our method significantly reduce the cost of the estimates. First, the Influence Graph  $G_I$  is created *without* observing the sample data. The mutation data and  $G_I$  are then combined to create the sample dependent graphs  $G_I(\delta)$  and  $H(\delta)$ . Thus, the Monte Carlo simulation needs to run on the graph  $G_I$  which is significantly smaller than the original interaction network (in our data the original interaction network had 18796 nodes while the influence graph had only about 600 nodes). Second, our statistical test does not use the  $p$ -values of individual connected subgraphs/components but the  $p$ -value of the distribution of the number of connected subgraphs/components of a given size. Thus, for this test it is sufficient to estimate  $p$ -values that are a magnitude larger, and therefore require significantly fewer rounds of simulations. These features allowed us to compute the null distributions through Monte-Carlo simulations for the size of our data with no significant computational cost. For larger number of tested genes we can estimate the null hypothesis through analytical bounds.

### 3 Experimental Results

We applied our approach to analyze somatic mutation data from two recent studies. The first dataset is a collection of 453 somatic mutations identified in 601 tested genes from 91 glioblastoma multiforme (GBM) samples from The Cancer Genome Atlas [31]. In total, 223 genes were reported mutated in at least one sample. The second dataset is a collection of 1013 somatic mutations identified in 623 tested genes from 188 lung adenocarcinoma samples from the Tumor Sequencing Project [8]. In total, 356 genes were reported mutated in at least one sample. For the Enhanced Influence model we also considered simulated data.

We use the protein interaction network from the Human Protein Reference Database (June 2008 version) [22] which consists of 18796 vertices and 37107 edges. We derive the influence graph for each dataset by directly computing the inverse [2] of  $L_\gamma$ . The results presented below are obtained by fixing the parameter  $\gamma = 8$ , which is approximately the average degree of a node in HPRD (after the

<sup>2</sup> In contrast [33] derive a power series approximation to  $L_\gamma^{-1}$  whose convergence depends on the choice of  $\gamma$ .

removal of disconnected nodes). We also considered  $\gamma = 1$  and  $\gamma = 30$ : in both cases the results obtained are close to the ones obtained with  $\gamma = 8$ .

The resulting influence graphs have weights  $i(g_j, g_k) \neq 0$  for almost all pairs  $(g_j, g_k)$  of tested genes: less than 2% of the weights are zero in the GBM graph, while all weights in the lung adenocarcinoma graph are positive. Supplementary tables are available at <http://www.cs.brown.edu/people/braphael/supplements/>.

### 3.1 Combinatorial Model

We used the combinatorial model to extract a subnetwork of  $k$  mutated genes that is mutated in the highest number of samples from GBM and lung adenocarcinoma with  $k = 10$  and  $k = 20$ . For both datasets we used the procedure described in Section 2.3 to derive the threshold  $\delta = 0.0001$  for the reduced influence graph  $G_I(\delta)$ . Table 1 shows that we find statistically significant subnetworks under both the  $H_0^{\text{gene}}$  and  $H_0^{\text{sample}}$  null hypotheses ( $p$ -values for  $H_0^{\text{sample}}$  are computed without Monte-Carlo simulation). To assess the biological significance of our findings in GBM, we compared the genes in each subnetwork to the genes in pathways that were previously implicated in GBM and used as a benchmark in the TCGA publication [31] (See also Figure 2 (a) below). We find that our subnetworks are enriched for genes in the RTK/RAS/PI(3)K pathway and to a lesser extent, the p53 pathway. For the lung adenocarcinoma samples, we find that the subnetworks share significant overlap with the pathways reported in the original publication [8]. These results demonstrate that the combinatorial model is effective in recovering genes known to be important in each of these cancers.

### 3.2 Enhanced Influence Model

*Simulated Data.* We tested the ability of our enhanced influence model to recover significantly mutated pathways in simulated data. We extracted a well-curated network of 258 genes called “Pathways in cancer (hsa05200)” from the KEGG database [20]. We augmented this network with additional random edges so that 20% of the edges of the resulting network were random. We assigned mutations to a well-known cancer signaling pathway, PKC -RAF - MEK - ERK, a linear chain  $\mathcal{P}$  of 4 genes, so that at least one gene is mutated in  $x\%$  of samples, for different  $x$ . We then randomly assigned mutations to all the genes in the network matching the observed values (e.g. number of samples, ratio between number of tested genes and number of genes in the network, etc.) We correctly identify  $\mathcal{P}$  as significantly mutated ( $P < 10^{-2}$ , FDR  $< 10^{-2}$ ) even when each gene in  $\mathcal{P}$  is altered in  $\leq 5\%$  of the samples, but  $\mathcal{P}$  is altered in 17% of the samples. Note that genes mutated in 5% of the samples were not reported as significantly mutated in [31], demonstrating that our method correctly identifies a mutated path even when the individual genes in the path are not mutated in a significant number of samples. Moreover,  $\mathcal{P}$  is the *only* significant pathway reported by our method. To verify that our influence measure takes into account the topology of the network, we added a number of edges to the RAF gene in  $\mathcal{P}$ , giving it high

**Table 1.** Results of the combinatorial model.  $k$  is the number of genes in the subnetwork.  $samples$  is the number of samples in which the subnetwork is mutated.  $p$ -val is the probability of observing a connected subgraph of size  $k$  under the random model  $H_0^{sample}$  or  $H_0^{gene}$ . *enrichment p-val* is the  $p$ -value of the hypergeometric test for overlap between genes in the identified subgraph and genes reported significant pathways in [31] or [8]. For GBM, *enrichment p-val* is the  $p$ -value of the hypergeometric test for RTK/RAS/PI(3)K and p53 pathways.

dataset	$k$	samples	p-val		pathway enrichment p-val		
			$H_0^{sample}$	$H_0^{gene}$	all	RTK/RAS/PI(3)K	p53
GBM	10	67	$< 10^{-10}$	$4 \times 10^{-3}$	$3 \times 10^{-4}$	$8 \times 10^{-4}$	0.19
	20	78	$< 10^{-10}$	$< 10^{-3}$	$10^{-5}$	$8 \times 10^{-5}$	0.05
Lung	10	140	$< 10^{-10}$	0.02	$8 \times 10^{-6}$	/	
	20	151	$< 10^{-10}$	0.03	$3 \times 10^{-3}$	/	

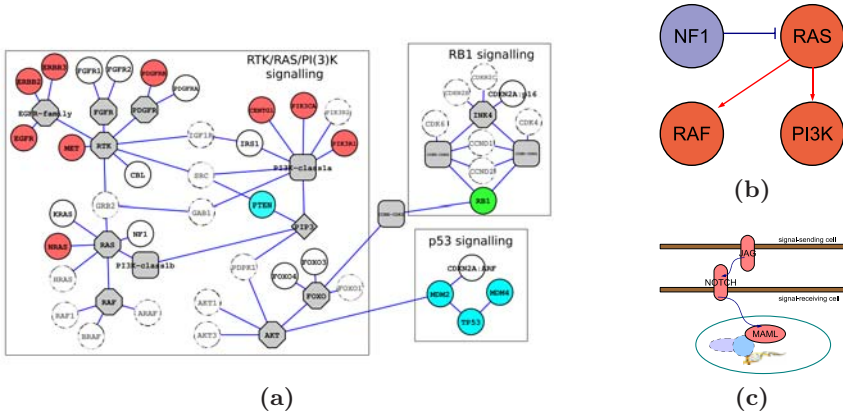
degree in the network. As expected,  $\mathcal{P}$  is no longer identified as significant in the modified network.

*Real data.* We applied the enhanced influence model to the GBM and lung adenocarcinoma datasets. Following the procedure described in Section 2.4, we first computed the enhanced influence network, using a threshold of  $\delta = 0.003$  for the GBM data and  $\delta = 0.01$  for the lung adenocarcinoma data. Table 2 shows the number and sizes of the connected components identified in the GBM data, and the associated  $p$ -values, the latter obtained using the method described in Section 2.5. We identify two significant connected components with more than 19 genes ( $FDR \leq 0.14$ ). We find significant overlap ( $P < 10^{-2}$  by hypergeometric test) between the 68 genes in our connected components and the set of all mutated genes in the same RTK/RAS/PI(3)K, p53, and RB pathways examined in the TCGA study [31]. The second largest connected component with 19 genes has significant overlap to the p53 pathway, while the largest connected component with 22 genes has significant overlap with the RTK/RAS/PI(3)K signaling pathway. In contrast to the combinatorial model, the enhanced influence model separates these two pathways into different connected components. Figure 2 (a) illustrates the overlap between the mutated genes in connected components returned by our method and genes in the pathways reported in [31].

For the lung data, Table 3 shows the sizes of connected components returned by the enhanced influence model and the  $p$ -values associated with each. The 88 genes in the union of the connected components derived by our method overlap significantly ( $P < 7 \times 10^{-9}$  by the hypergeometric test) with the mutated pathways reported in the network of Figure 6 in the TSP publication [8]. We identify 4 connected components of size  $\geq 7$  ( $FDR \leq 0.56$ ). The first connected component of size 10 contains genes in the p53 pathway, and the second one is enriched ( $P < 10^{-2}$ ) for the MAPK pathway (Figure 2 (b)). The third component is the ephrin receptor gene family, a large family of membrane-bound receptor tyrosine kinases, that were reported as mutated in breast and colorectal cancers [36]. Notably, only one of the genes in this component, EPHA3, is mentioned as

**Table 2.** Results of the enhanced influence model on GBM samples.  $s$  is the size of connected components (c.c.) found with our method.  $\# c.c. \geq s$  is the number of c.c. with at least  $s$  nodes.  $\mu$  is the expected number of c.c. with  $\geq s$  nodes under random models  $H_0^{\text{gene}}$ ,  $H_0^{\text{sample}}$ .  $p\text{-val}$  is the probability of observing at least  $\# c.c. \geq s$  with at least  $s$  nodes in a random dataset. The last 3 columns show, for c.c. with  $s > 3$ , the result of the hypergeometric test for enrichment for RTK/RAS/PI(3)K, and p53 pathways respectively.

$s$	$\# c.c. \geq s$	$H_0^{\text{sample}}$		$H_0^{\text{gene}}$		enrichment p-val	
		$\mu$	p-val	$\mu$	p-val	RTK/RAS/PI(3)K	p53
2	15	22.18	0.97	13.63	0.38	/	/
3	3	6.37	0.98	4.38	0.6	/	/
19	2	$< 10^{-3}$	$< 10^{-3}$	0.07	$< 10^{-3}$	0.9	$4 \times 10^{-3}$
22	1	$< 10^{-3}$	$< 10^{-3}$	0.05	0.05	$4 \times 10^{-6}$	-



**Fig. 2.** (a) Overlap between subnetworks found by the enhanced influence model and significant pathways reported in [31]. Each circle is a gene, gray nodes represents protein families or complexes, or small molecules. For each protein family and complex, tested genes are shown. “Dashed” nodes are tested genes that were not mutated in GBM, and thus cannot be returned as significant. Red nodes are found in the c.c. of size 22, blue nodes in the c.c. of size 18, and the green node in a c.c. of size 2. (b) Pathway corresponding to one of the connected components extracted with enhanced influence model in lung. (c) Notch signaling pathway identified in the lung dataset.

significantly mutated in [8]. Finally, the connected component of size 7 consists exclusively of members of the Notch signaling pathway (Figure 2(c)). The mutated genes include: the Notch receptor (NOTCH2/3/4); Jagged (JAG1/2), the ligand of Notch; and Mastermind (MAML1/2), a transcriptional co-activator of Notch target genes. The Notch signaling pathway is a major developmental pathway that has been implicated in a variety of cancers [1] including lung cancer [7]. Mutations in this pathway were not noted in the original TSP publication [8], probably because no single gene in this pathway is mutated in more than



**Table 3.** Results of the enhanced influence model on lung adenocarcinoma samples. Columns are as described in Table 2. Last column shows, for c.c. with  $s \geq 7$ , the result of the hypergeometric test for enrichment all genes reported in significant pathways in 8 (the 3 values shown refers to c.c. of size 10).

$s$	# c.c. $\geq s$	$H_0^{\text{sample}}$		$H_0^{\text{gene}}$		enrichment	p-val
		$\mu$	p-val	$\mu$	p-val		
2	24	23.4	0.7	17.67	0.4	/	
3	11	6.51	0.13	7.27	0.2	/	
4	7	3.21	0.07	4.98	0.13	/	
5	5	2.09	0.01	2.18	0.01	/	
7	4	0.54	0.01	0.56	0.01	-	
10	3	$< 10^{-3}$	$< 10^{-3}$	0.4	0.02	0.34; $10^{-5}$ ; $9 \times 10^{-8}$	

3 samples. Because our method exploits both mutation frequency and network topology, we are able to identify these more subtle mutated pathways, and in this case identify an entire “signaling” circuit.

### 3.3 Naïve Approach

To demonstrate the impact of the influence graph on the results, we implemented a naïve approach that examines all paths in the original HPRD network that connect two tested genes and contain at most 3 nodes. We extracted all paths that were altered in a significant number of samples with  $\text{FDR} \leq 0.01$  using the standard Benjamini-Yekutieli method [4]. More than 1700 paths in GBM and  $> 2200$  in lung adenocarcinoma are marked as significant with this method. A major reason for this large number of paths is the presence of highly mutated genes that are also high-degree nodes in the HPRD network (e.g. TP53). Each path through these high degree nodes is marked as significant. One possible solution is to remove any path that contains a subpath that is significant. However, these filtered paths include *none* through important highly-mutated and high degree genes (like TP53). Our influence graph uses both mutation frequency and local topology of the network, allowing us to recover subnetworks containing these genes. Finally, we note that finding larger, statistically significant subnetworks (e.g. those with 10 or 20 nodes) with the naïve approach is impossible in the GBM and lung datasets because of the severe multiple hypotheses correction for the large number of subnetworks tested; e.g., the number of connected components with 10 tested nodes in the HPRD network is  $> 10^{10}$ . For the same reason the enumeration of all the paths or connected components of reasonable size is impossible.

## 4 Discussion

We present an approach to identify significantly mutated pathways in a large, unannotated interaction network. The subnetworks derived by our method share



significant overlap with the known cancer pathways such as the manually curated pathways in TCGA [31]. Remarkably, we automatically extracted a large fraction of these pathways with modest number (100-200) of samples (Figure 2). Our approach has two key advantages over the common strategy of testing the overlap between mutated genes and genes from known pathways approach, using a hypergeometric or similar test. First, we incorporate biological information that is not presently represented in existing well-characterized pathways, while accounting for the uncertainty in large gene interaction networks. Second, we are able to assign significance to genes that are altered at low frequency but are part of a larger subnetwork that is altered at significant frequency. The latter advantage was demonstrated in the lung adenocarcinoma dataset where we identify the Notch signaling pathway as significant, even though the individual genes were not mutated at significant frequency.

We plan to extend our model in numerous directions, including: (i) inclusion of other types of mutations such as copy number changes in genes, genome rearrangements, gene expression, or epigenetic alterations; (ii) extension of the interaction network to include additional interaction types (e.g. regulatory or miRNA) as well as directed interactions (activating vs. inhibitory); (iii) consideration of errors in the interaction network. The later can be included naturally in our diffusion model by adding weights, or reliabilities, on the edges. Moreover, we have adapted our model to take into account the length of the genes in the network, weighting the frequency of mutation in a gene by its length. The results obtained for the GBM and lung adenocarcinoma data are extremely close to the one presented here (data not shown).

We anticipate that our method will become even more useful as larger datasets become available. Several recent studies [40,18,32] have surveyed a much larger number of genes than considered here (approximately 20,000), but in a relatively small number of samples (1-2 dozen per cancer type). Continuing decline in sequencing costs and the development of targeted exon-capture techniques [15] will soon enable global surveys of all protein-coding genes in hundreds to thousands of cancer samples.

## References

1. Axelson, H.: Notch signaling and cancer: emerging complexity. *Semin. Cancer Biol.* 14, 317–319 (2004)
2. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F., Pawson, T., Hogue, C.W.: BIND—The Biomolecular Interaction Network Database. *Nucleic Acids Res.* 29, 242–245 (2001)
3. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate. *J. Royal Statistical Society, Series B* 57, 289–300 (1995)
4. Benjamini, Y., Yekutieli, D.: The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics* 29(4), 1165–1188 (2001)
5. Chuang, H.Y., Lee, E., Liu, Y.T., Lee, D., Ideker, T.: Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* 3, 140 (2007)
6. Chung, F.: The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences* 104(50), 19735 (2007)

7. Collins, B.J., Kleeberger, W., Ball, D.W.: Notch in lung development and lung cancer. *Semin. Cancer Biol.* 14, 357–364 (2004)
8. Ding, L., et al.: Somatic mutations affect key pathways in lung adenocarcinoma. *Nature* 455(7216), 1069–1075 (2008)
9. Doyle, P.G., Snell, J.L.: *Random Walks and Electric Networks*. The Mathematical Association of America (1984)
10. Feige, U., Kortsarz, G., Peleg, D.: The dense k-subgraph problem. *Algorithmica* 29, 2001 (1999)
11. Greenman, C., et al.: Patterns of somatic mutation in human cancer genomes. *Nature* 446, 153–158 (2007)
12. Hahn, W.C., Weinberg, R.A.: Modelling the molecular circuitry of cancer. *Nat. Rev. Cancer* 2(5), 331–341 (2002)
13. Hescott, B.J., Leiserson, M.D.M., Cowen, L.J., Slonim, D.K.: Evaluating between-pathway models with expression data. In: Batzoglou, S. (ed.) *RECOMB 2009*. LNCS, vol. 5541, pp. 372–385. Springer, Heidelberg (2009)
14. Hochbaum, D.S. (ed.): *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston (1997)
15. Hodges, E., et al.: Genome-wide in situ exon capture for selective resequencing. *Nat. Genet.* 39, 1522–1527 (2007)
16. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* 18(suppl. 1), S233–S240
17. Jensen, L.J., et al.: STRING 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* 37, D412–D416 (2009)
18. Jones, S., et al.: Core signaling pathways in human pancreatic cancers revealed by global genomic analyses. *Science* 321(5897), 1801–1806 (2008)
19. Jonsson, P.F., Bates, P.A.: Global topological features of cancer proteins in the human interactome. *Bioinformatics* 22, 2291–2297 (2006)
20. Kanehisa, M., Goto, S.: KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 28, 27–30 (2000)
21. Karni, S., Soreq, H., Sharan, R.: A network-based method for predicting disease-causing genes. *J. Comput. Biol.* 16, 181–189 (2009)
22. Keshava Prasad, T.S., et al.: Human Protein Reference Database–2009 update. *Nucleic Acids Res.* 37, D767–D772 (2009)
23. Kirsch, A., Mitzenmacher, M., Pietracaprina, A., Pucci, G., Upfal, E., Vandin, F.: An efficient rigorous approach for identifying statistically significant frequent itemsets. In: *PODS*, pp. 117–126 (2009)
24. Kondor, R.I., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: *Proceedings of the ICML*, pp. 315–322 (2002)
25. Lin, J., et al.: A multidimensional analysis of genes mutated in breast and colorectal cancers. *Genome Res.* 17, 1304–1318 (2007)
26. Liu, M., et al.: Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genet.* 3, e96 (2007)
27. Lovász, L.: *Random walks on graphs: A survey* (1993)
28. Ma, X., Lee, H., Wang, L., Sun, F.: CGI: a new approach for prioritizing genes by combining gene expression and protein-protein interaction data. *Bioinformatics* 23, 215–221 (2007)
29. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21(suppl. 1), i302–i310 (2005)

30. Nacu, S., Critchley-Thorne, R., Lee, P., Holmes, S.: Gene expression network analysis and applications to immunology. *Bioinformatics* 23, 850–858 (2007)
31. The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 455(7216), 1061–1068 (2008)
32. Parsons, D.W., et al.: An integrated genomic analysis of human glioblastoma multiforme. *Science* 321(5897), 1807–1812 (2008)
33. Qi, Y., Suhail, Y., Lin, Y.Y., Boeke, J.D., Bader, J.S.: Finding friends and enemies in an enemies-only network: a graph diffusion kernel for predicting novel genetic interactions and co-complex membership from yeast genetic interactions. *Genome Res.* 18, 1991–2004 (2008)
34. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res.* 32, D449–D451 (2004)
35. Shuai, T.-P., Hu, X.: Connected set cover problem and its applications. In: Cheng, S.-W., Poon, C.K. (eds.) *AAIM 2006*. LNCS, vol. 4041, pp. 243–254. Springer, Heidelberg (2006)
36. Sjoblom, T., et al.: The consensus coding sequences of human breast and colorectal cancers. *Science* 314(5797), 268–274 (2006)
37. Tsuda, K., Noble, W.S.: Learning kernels from biological networks by maximizing entropy. *Bioinformatics* 20(suppl. 1), i326–i333 (2004)
38. Ulitsky, I., Karp, R.M., Shamir, R.: Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In: Vingron, M., Wong, L. (eds.) *RECOMB 2008*. LNCS (LNBI), vol. 4955, pp. 347–359. Springer, Heidelberg (2008)
39. Vogelstein, B., Kinzler, K.W.: Cancer genes and the pathways they control. *Nat. Med.* 10, 789–799 (2004)
40. Wood, L.D., et al.: The genomic landscapes of human breast and colorectal cancers. *Science* 318(5853), 1108–1113 (2007)

# Leveraging Sequence Classification by Taxonomy-Based Multitask Learning

Christian Widmer<sup>1</sup>, Jose Leiva<sup>1,2</sup>, Yasemin Altun<sup>2</sup>, and Gunnar Rätsch<sup>1</sup>

<sup>1</sup> Friedrich Miescher Laboratory, Max Planck Society, Spemannstr. 39,  
72076 Tübingen, Germany

<sup>2</sup> Max Planck Institute for Biological Cybernetics, Spemannstr. 38,  
72076 Tübingen, Germany

**Abstract.** In this work we consider an inference task that biologists are very good at: deciphering biological processes by bringing together knowledge that has been obtained by experiments using various organisms, while respecting the differences and commonalities of these organisms. We look at this problem from an sequence analysis point of view, where we aim at solving the same classification task in different organisms. We investigate the challenge of combining information from several organisms, whereas we consider the relation between the organisms to be defined by a tree structure derived from their phylogeny. Multitask learning, a machine learning technique that recently received considerable attention, considers the problem of learning across tasks that are related to each other. We treat each organism as one task and present three novel multitask learning methods to handle situations in which the relationships among tasks can be described by a hierarchy. These algorithms are designed for large-scale applications and are therefore applicable to problems with a large number of training examples, which are frequently encountered in sequence analysis. We perform experimental analyses on synthetic data sets in order to illustrate the properties of our algorithms. Moreover, we consider a problem from genomic sequence analysis, namely splice site recognition, to illustrate the usefulness of our approach. We show that intelligently combining data from 15 eukaryotic organisms can indeed significantly improve the prediction performance compared to traditional learning approaches. On a broader perspective, we expect that algorithms like the ones presented in this work have the potential to complement and enrich the strategy of homology-based sequence analysis that are currently the quasi-standard in biological sequence analysis.

## 1 Introduction

Over a decade ago, an eight-year lasting collaborative effort resulted in the first completely sequenced genome of a multi-cellular organism, the free-living nematode *Caenorhabditis elegans*. Today, more than 50 eukaryotic genomes have been sequenced and several hundred more are underway. The genome sequences are the basis for much of the research on the molecular processes in these organisms.

Typically, the more closely related the organisms are, the more similar are these processes. For some organisms, certain biochemical experiments for the analysis of particular processes can be performed more readily than for others (i.e. a large part of biological understanding was obtained from experiments based on a few model organisms such as yeast). This understanding can then be transferred to other organisms, for instance by verifying or refining models of the processes—often at a fraction of the original cost. This is but one example of a situation, where transfer of knowledge across organisms is very fruitful.

In computational biology we often study the problem of building statistical models from data in order to predict, analyze, and ultimately understand biological systems. Regardless of the problem at hand, be it the recognition of sequence signals such as splice sites, the prediction of protein-protein interactions, or the modeling of metabolic networks, we frequently have access to data sets for multiple organisms. Thus, our goal is to develop methods that aim at taking advantage of the data from different organisms in order to improve the performance of the statistical models built for all organisms. We argue that, when building a predictor for a given organism, data from other organisms should be incorporated to the extent of the relation between the organisms.

Since it is assumed that all life can be traced back to an ancient common ancestor, all organisms can ultimately be related by phylogeny. Furthermore, if two organisms share a sufficiently long evolutionary history before divergence, it can be expected that certain biological mechanisms (e.g., splicing) are conserved to some degree. Thus, it is reasonable to assume that we can leverage data from other organisms to enhance model quality for the organism of interest. In bioinformatics, this is traditionally done by considering sequence homology. This approach, however, is limited to almost exact correspondences of sequences between one or several biological sequences, while it fails to capture other features such as sequence composition that can be used to build an accurate model.

A family of machine learning methods, commonly referred to as *domain adaptation* or *transfer learning*, investigates the application of a predictor trained with data from a given domain to data from a different one (see e.g., [35,11]). Furthermore, the so-called *multitask learning* techniques consider the problem of simultaneously obtaining predictors from different domains by exploiting the fact that the domains are related (see e.g., [16]). Most of these methods assume uniform relations across domains/tasks.<sup>1</sup> However, it is conceivable that sharing information between closely related domains is more beneficial than sharing between domains that are only distantly related (according to a given criterion). Hence, it is important to take into account the degree of relatedness among the domains when obtaining the set of models. Here, we investigate multitask learning scenarios where we are given *a priori* information about a hierarchy that relates the domains at hand, which is often the case for biological problems. In particular, we treat each organism as a domain and employ the hierarchy given by the phylogeny. The fact that the availability of data describing the same biological mechanism in several organisms is a reoccurring theme makes

---

<sup>1</sup> We use the terms *task* and *domain* interchangeably.

the hierarchical multitask learning approach particularly well suited for many applications in computational biology.

Building upon previous work [11], we propose a general framework for leveraging information from related organisms by ensuring correspondence on model basis rather than directly comparing sequences. We consider two principal approaches to incorporate relations across domains.

In the first approach, for a given task  $t$ , models from the other tasks serve as prior information to the model of  $t$ , such that the parameters  $\mathbf{w}_t$  of task  $t$  are close to the parameters  $\mathbf{w}_o$  of the other models. This can be achieved by minimizing the norm of the differences of the parameter vectors,  $\|\mathbf{w}_t - \mathbf{w}_o\|$ , along with the original loss function. A convenient way of implementing this approach is training models in a top-down manner, where a model is learned for each node in the hierarchy over the data sets of tasks spanned by the node. The parent nodes are used as the prior information,  $\|\mathbf{w}_t - \mathbf{w}_{parent(t)}\|$ . Here, one can readily use existing inference techniques with slight changes to the implementation. We describe this method in Section 2.2. Accordingly, one can use the models of all tasks as prior information,  $\|\mathbf{w}_t - \mathbf{w}_{t'}\|$  for related tasks  $t$  and  $t'$ , and train the parameters of all tasks jointly. This method is outlined in Section 2.3. Compared to the top-down approach, this formulation involves a larger set of parameters to be jointly optimized during training. However, it can be decomposed into sub-problems, which in turn are solved in an iterative manner. Its advantage is that each problem can be trained on smaller data sets compared to the top-down approach in which the model of the root node is trained on the union of all data sets.

An alternative to the latter pair-wise approach has been suggested in the context of support vector machines (SVMs) for the special case of two tasks [5]. We extend this idea and design an appropriate kernel function that not only considers the data of the task  $t$ , but also the data from all other tasks according to their similarities. We show that this kernel design can be derived by defining predictor functions over the task parameters as well as the parameters of the ancestors of the task. This leads to an essentially effortless multitask approach, since one can use standard SVM implementations by simply implementing the new kernel. We describe the new kernel and its derivation in Section 2.4.

In Section 3 we evaluate the proposed algorithms on simulated data and illustrate some of their properties. Moreover, we consider the problem of splice-site recognition in 15 different eukaryotic genomes and show that the proposed methods can significantly improve the prediction accuracy by combining the information available for all organisms. We conclude the paper with a discussion in Section 4.

## 2 Hierarchical Multitask Learning

### 2.1 Preliminaries

We are interested in the problem of learning  $M$  functions  $f_t : \mathcal{X} \rightarrow \mathcal{Y}$  between the input space  $\mathcal{X}$  and discrete output space  $\mathcal{Y}$ , where  $t = 1, \dots, M$  corresponds

to a task. We assume that we are given the relations across tasks via a hierarchy  $\mathcal{T}$ , where the tasks are the leaves of  $\mathcal{T}$ . Our goal is to make use of the training sample  $S_t$  of task  $t$ , while also considering the training samples from the other tasks according to the given hierarchical information.

We investigate two principal approaches for exploiting hierarchical information about task relations in a multitask learning (MTL) framework; namely incorporating prior knowledge via *regularization* and via *kernel design*. The first approach is used in two of the proposed methods. Hence, we describe the underlying idea before we go into the technical detail. A regularization term is typically used to introduce a penalty for complex solutions into the optimization problem of a learning algorithm. In the existence of prior knowledge, the Empirical Risk Minimization framework [13] incorporates the prior knowledge  $\bar{f}$  by

$$\hat{f} = \min_f \left[ R(f - \bar{f}) + \sum_{(\mathbf{x}, y) \in S} \ell(f(\mathbf{x}), y) \right], \quad (1)$$

where  $R$  is the regularization term that penalizes the deviation of the current model  $f$  from the previously obtained (fix) model  $\bar{f}$ , and  $\ell$  is a loss function (such as the squared loss, or the hinge loss) that penalizes the error on the training sample  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ . In the multitask approach, we use the models of related nodes as  $\bar{f}$ , where relations are given by  $\mathcal{T}$ . Following this scheme, any regularized Machine Learning framework (e.g., regularized least squares, regularized logistic regression) can be extended to include prior information.

Since one of the main goals of this work is to provide learning algorithms that scale to large amounts of data, we instantiate the concept for Support Vector Machines (SVM) [13, 2]. It has been shown in previous work that SVMs using string kernels such as the Spectrum [8] or the Weighted Degree Kernel (WDK) [9] are well suited for nucleic and protein sequence analysis [2].

## 2.2 Top-Down Domain Adaptive Support Vector Machines

Our first approach uses a regularized multitask approach, where a model is learned for each node of the hierarchy  $\mathcal{T}$  and the parent of node  $v$  serves as prior information to  $v$ , such that the final model of  $v$  is close to the model of its parent. The idea is to train the models in a top-down fashion, where the most general model is obtained at the root node and more domain specific models are obtained by moving down toward the leaves.

In SVMs, a model  $f$  is a linear function parametrized by  $\mathbf{w}$  and  $b$ ,  $f = \langle \mathbf{x}, \mathbf{w} \rangle + b$ . Since each model can be trained independently, we drop the indices for tasks and simply use  $\mathbf{w}$  for parameters of the current node, and  $\mathbf{w}_0$  for the parameters of the parent node. The primal of the extended SVM formulation is

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_0\|^2 + C \sum_{(\mathbf{x}, y) \in S} \ell(\langle \mathbf{x}, \mathbf{w} \rangle + b, y), \quad (2)$$

<sup>2</sup> Other options, however, may also be suitable to implement the ideas of this work.

where  $\ell$  is the hinge loss,  $\ell(z, y) = \max\{1 - yz, 0\}$ . From a biological perspective, the penalty term  $\|\mathbf{w} - \mathbf{w}_0\|^2$  enforces the model of an organism and the organism it is derived from to be similar, based on the assumption of a relatively small mutation rate. If the latter is the case, most properties of the model are conserved through evolution. In order to employ kernels, we derive the dual of the above formulation:

$$\begin{aligned} \max_{\alpha} & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^n \alpha_i \underbrace{\left( \sum_{j=1}^m \alpha'_j y_i y'_j k(\mathbf{x}_i, \mathbf{x}'_j) \right)}_{p_i} - 1, \\ \text{s.t. } & \alpha^T \mathbf{y} = 0, \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, n\}, \end{aligned}$$

where  $n$  and  $m$  are the number of training examples in  $S$  and  $S_0$  respectively, and  $S_0$  is the training sample of the prior model. The  $\alpha_i$  represent the dual variables of the current learning problem, whereas the  $\alpha'_j$  represent the dual variables obtained from the prior model  $\mathbf{w}_0$ ; in the case of the linear kernel, it is described as  $\mathbf{w}_0 = \sum_{j=1}^m \alpha'_j y'_j \mathbf{x}'_j$ . The resulting prediction is performed by

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^m \alpha'_j y'_j k(\mathbf{x}, \mathbf{x}_j) + b.$$

In the dual form of the standard SVM, the term  $\mathbf{p}$  (corresponding to the  $p_i$  in the equation above) is set to  $\mathbf{p} = (-1, \dots, -1)^T$ . This is equivalent to the case, where we have no prior model (i.e.  $\mathbf{w}_0 = (0, \dots, 0)^T$ ). In our extended formulation, the  $p_i$  can be pre-computed and passed to the underlying SVM-solver as the linear term of the corresponding quadratic program (QP). To provide implementations that readily deal with large-scale learning problems, we have extended the SVM implementations *LibSVM* [4] and *SVMLight* [7] to handle prior information.<sup>3</sup>

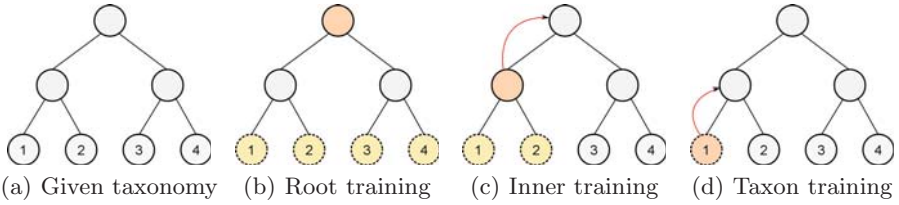
**Top-Down Hierarchical Learning.** An illustration of the training procedure is given in Figure 1. In a top-down manner, a predictor  $f$  is obtained at each node  $v$ , whereas the loss  $L$  is evaluated on the union of training data  $S = \cup\{S_{t \preceq \tau v}\}$  at the leaves below the current node  $v$ , while the current model  $f$  is regularized against the parent predictor  $f_p$ . Training is completed once we have obtained a predictor  $f_t$  for each task  $t$ . The algorithm will be referred to as *Top-Down-SVM* in the following.

### 2.3 Support Vector Machines with Pairwise Task Regularization

In the previous section, we described a method that learns models for internal nodes of the hierarchy and imposes regularization between a node and its parent. In this section, we describe a method where the relation between tasks is modeled

<sup>3</sup> <http://www.shogun-toolbox.org>





**Fig. 1.** Illustration of the hierarchical top-down MTL training procedure. In this example, we consider four tasks related via the tree structure in [1\(a\)](#). Each leaf is associated with a task  $t$  and its training sample  $S_t = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ . In [1\(b\)](#), training begins by obtaining the predictor at the root node, for which data from all leaves are taken into account in the loss term. Next, we move down one level to train a classifier at an inner node, as shown in [1\(c\)](#). Here, the loss is measured w.r.t.  $S_1 \cup S_2$  and the classifier is forced to be similar to the parent solution via the regularization term, as indicated by the red arrow. Finally, in [1\(d\)](#), we obtain the final classifier of task 1 by only taking into account  $S_1$  to measure the loss, while again regularizing against the parent predictor. The procedure is applied in a top-down manner to the remaining nodes until a predictor for each leaf was obtained.

directly via pairwise regularization. We refer to this method as *Pairwise* learning. Similar regularization-based multitask learning methods were proposed in [6](#). However, we extend this previous approach by incorporating hierarchical task information, and by providing an efficient decomposition for practicability.

Given the hierarchy  $\mathcal{T}$ , we first compute the distance  $d_{s,t}$  between the tasks  $s$  and  $t$  by summing up the lengths of the edges on the path from one leaf to the other (tree-hop-distance). In our experiments, we assumed all edge lengths to be 1. The resulting distance  $d_{s,t}$  is subsequently converted to a similarity  $\gamma_{s,t}$  by the transformation  $\gamma_{s,t} = a - d_{s,t}/d_{max}$ , where  $a \geq 1$  is a parameter to control the base similarity between tasks, and  $d_{max}$  is the maximal distance between any pair of leaves in  $\mathcal{T}$ . The matrix  $\Gamma$  that captures all pairwise similarities  $\gamma_{s,t}$  will be referred to as task similarity matrix in the following.

In this method, the *prior* information comes from the models of similar tasks. All the models are trained jointly by optimizing the regularization term along with the loss  $\ell$ , which is measured separately for each  $\mathbf{w}_t$  on the corresponding data set  $S_t$ ,

$$\min_{\mathbf{w}_1, \dots, \mathbf{w}_M} \frac{1}{2} \sum_{t=1}^M \sum_{s=1}^M \gamma_{s,t} \|\mathbf{w}_t - \mathbf{w}_s\|^2 + \sum_{t=1}^M C_t \sum_{(\mathbf{x}, y) \in S_t} \ell(\langle \mathbf{x}, \mathbf{w}_t \rangle, y).$$

The parameter  $C_t$  is used to trade off the loss for task  $t$  on the training sample against the regularization term in order to control generalization performance. In our experiments, we set  $C_t = C$ , for  $t = 1, \dots, M$  for simplicity. The biological interpretation of this formulation is that if two organisms share a sufficiently long evolutionary history before divergence (reflected by  $\gamma_{s,t}$ ), it can be expected that certain aspects of the model describing the biological mechanism are conserved. Thus, we use the task-similarity matrix  $\Gamma$  to control how strongly we regularize each  $(\mathbf{w}_t, \mathbf{w}_s)$  pair to be close to each other.

**Decomposition.** The pairwise formulation learns models only for the leaf nodes of the hierarchy, as opposed to the *Top-Down-SVM* approach, in which the number of models to learn is given by all the nodes of the hierarchy. Its comparative disadvantage, on the other hand, is that it couples the parameters of all models, which leads to a large optimization problem. In order to overcome this limitation, we developed a decomposition of the optimization problem that allows the global solution to be obtained by solving a series of SVM-like quadratic programs iteratively until convergence. It can be shown that the above optimization problem has a fixed point that coincides with the optimization problem of

$$\min_{\mathbf{w}_t} \frac{1}{2} \lambda_t \|\mathbf{w}_t - \mathbf{r}_t\|^2 + C \sum_{(\mathbf{x}, y) \in S_t} \ell(\langle \mathbf{x}, \mathbf{w}_t \rangle, y), \quad \forall t \tag{3}$$

$$\begin{aligned} \mathbf{r}_t &= \sum_{s \neq t} \beta_{ts} \mathbf{w}_s, \\ \lambda_t &= \sum_{s \neq t} \gamma_{ts}, \quad \beta_{ts} = \gamma_{ts} / \lambda_t. \end{aligned} \tag{4}$$

This formulation decouples the optimization problem into individual tasks and, hence, retains scalability. It states that the regularization prior  $\mathbf{r}_t$  of each task should be in the convex hull of  $\{\mathbf{w}_s\}_{s \neq t}$ . It can be solved iteratively by finding the optimal  $\mathbf{r}_t$  for the current  $\mathbf{w}_t$  (cf. (4)) and finding the optimal  $\mathbf{w}_t$  for the current  $\mathbf{r}_t$  (cf. (3)) for each task until convergence. Note that the difference between (3) and (2) is the prior model, where in the first case it is the weight average of related tasks and in the latter it is the parent model. Hence, the SVM implementations in Section 2.2 can be used to solve (3). Kernelization follows similarly. Investigating the relation between the dual and primal parameters,

$$\mathbf{w}_t = \sum_{i: \mathbf{x}_i \in S_t} \alpha_i y_i \mathbf{x}_i + \sum_{s \neq t} \beta_{ts} \sum_{j: \mathbf{x}_j \in S_s} \alpha_j y_j \mathbf{x}_j,$$

reveals that the pairwise regularization method results in “borrowing” support vectors of related models with respect to the task similarities.

### 2.4 Multitask-Kernel Learning

In Sections 2.2 and 2.3, we presented two hierarchical multitask approaches based on regularization with respect to related models. In this section, we propose an alternative approach, *MultiKernel*, by defining a kernel function that incorporates data from related tasks. We first define a matrix  $\Lambda$  to encode ancestry relationships as follows: Let  $\lambda_{tr}$  be the similarity of a task  $t$  and an inner node  $r$ . As in the previous section,  $\lambda_{tr}$  is inversely related to the distance between  $t$  and  $r$ , if  $r$  is an ancestor of  $t$  with respect to  $\mathcal{T}$ ,  $t \preceq_{\mathcal{T}} r$ , and 0 otherwise. We then define the predictor function for task  $t$  as

$$f_t(\mathbf{x}) = \mathbf{w}_t^T \mathbf{x} + b_t = (\mathbf{u}_t + \sum_{t \preceq r} \lambda_{tr} \mathbf{v}_r)^T \mathbf{x} + b_t,$$

where  $\{\mathbf{v}\}_r$  are the internal node parameters and  $\mathbf{u}_t$  are the leaf node (task) parameters. Here, the parameters of the node at each level in the hierarchy

represent the corresponding level of abstraction. More precisely, the parameters of the root node capture the most common structure across all tasks and as we descend in the hierarchy, the parameters capture the deviation from the previous level. Our goal is to achieve the proper specialization level for each task by combining these parameters in the prediction function.

We propose to obtain  $\{\mathbf{u}_t\}$  and  $\{\mathbf{v}_r\}$  by solving the regularized loss function for all tasks,

$$\min_{\{\mathbf{u}_t\},\{\mathbf{v}_r\}} \frac{1}{2} \sum_{t=1}^M \|\mathbf{u}_t\|^2 + \frac{1}{2} \sum_{r=1}^R \|\mathbf{v}_r\|^2 + C \sum_{t=1}^M \sum_{(\mathbf{x},y) \in S_t} \ell(\langle \mathbf{x}, \mathbf{w}_t \rangle, y),$$

where  $\ell$  is the hinge loss, and  $R$  is the number of inner nodes. Note that the tasks are related to each other through the internal node parameters  $\mathbf{v}$  as in the case of *Top-Down-SVM*. However, the loss term  $\ell$  is evaluated only on the leaf nodes, as opposed to *Top-Down-SVM*, where  $\ell$  is evaluated at all nodes in the hierarchy by combining the relevant data sets. Instead of learning the internal node models by error minimization and then enforcing the models of the parent-child nodes to be similar, the goal here is to learn the internal node models directly by minimizing the error on the leaf nodes (i.e. the tasks). It can be shown that the dual formulation of the problem above is equivalent to that of the standard SVM

$$\begin{aligned} \max_{\alpha} & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \tilde{k}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^n \alpha_i \\ \text{s.t.} & \quad \alpha^T \mathbf{y} = 0, \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, n\}, \end{aligned}$$

where the kernel is defined on the union of all data sets,

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = \tilde{\gamma}_{t(i),t(j)} k(\mathbf{x}_i, \mathbf{x}_j).$$

Here,  $t(i)$  denotes the task that data point  $\mathbf{x}_i$  belongs to, and  $\tilde{\gamma}_{ts}$  are the entries of  $\tilde{\mathbf{\Gamma}} = \mathbf{I} + \Delta \Delta^T$ . We require  $\tilde{\mathbf{\Gamma}}$  to be positive semi-definite in order to yield a valid kernel  $\tilde{k}$ . Hence, we have constructed a kernel  $\tilde{k}$  that incorporates the interaction among tasks in addition to the interaction among data points. This formulation is a generalization of the domain adaptation method of [5], where the reweighting of the original kernel matrix is less flexible. Lastly, please note that, while we needed  $\Delta$  for the sake of derivation, it becomes clear from the dual formulation that  $\tilde{\mathbf{\Gamma}}$  may be obtained using the same transformation as described in Section 2.3, which was done for the following experiments.

### 3 Results

*Experimental Setup.* We performed experiments on two types of data sets. First, we considered synthetic sequence data, which was created by applying mutations to a Position Specific Scoring Matrix (PSSM) according to a pre-defined binary

tree structure. Balanced, equally sized training data sets, 100 examples each, were sampled for each of the leaves. As test set, an additional 5000 examples were sampled for each task. We used the area under the ROC curve to evaluate the prediction performance. (An evaluation using the area under the precision recall curve yields similar results.)

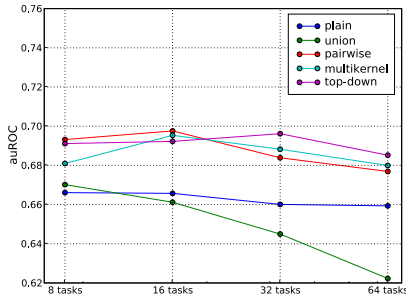
As a second application, we considered the problem of splice site recognition. We generated and used labeled sequences of acceptor splice sites from 15 different eukaryotic genomes, which were similarly generated in [10,11] (see Figure 3 for the taxonomic relation between the organisms). For each task, we obtained 10000 training examples and an additional test set of 5000 examples. We normalized the data sets to have 100 negative examples per positive example. We report the area under the precision recall curve (auPRC), which is an appropriate measure for unbalanced classification problems (i.e. detection problems) [9].

Experiments were performed for each of the presented MTL methods (*Top-Down-SVM*, *Pairwise*, *MultiKernel*) and the following two baseline methods. In *Union*, all data are combined into one data set  $S = \cup_{t=1}^M S_t$ , and a single global model is obtained, which is used to predict on all domains. Furthermore, we consider the baseline method *Plain*, in which an individual SVM is trained on the data  $S_i$  of each domain separately, not taking into account any information from the other domains. For each method, the regularization parameter  $C$  was chosen by cross-validation with 4 and 3 splits for toy and splice data, respectively. After obtaining the optimal  $C$ , we retrained the model on all available training data. The performance was measured on the separate test sets, which are considered large enough to obtain reliable estimates of the predictors' performances. The data sets, as well as the Appendix with more detail about hyper-parameter selection and toy data generation will be made available on our web-site [4].

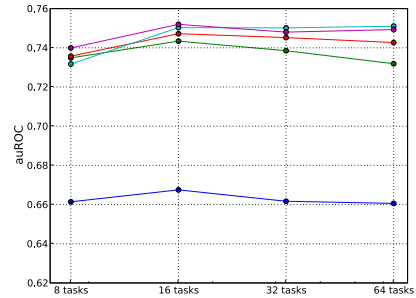
*Results on the Toy Datasets.* We consider two different settings of generating the toy data sets: one with relatively high mutation rate (larger differences between neighbouring tasks in the hierarchy) and one with small mutation rate (all tasks are closely related). For this study, we analyze how the different methods perform in these two settings with respect to the number of tasks (8, 16, 32, 64 tasks). The results are shown in Figure 2. We can observe that the two baseline methods perform very differently: While the *Plain* method performs essentially indifferent for different numbers of tasks and mutation rates, the *Union* method performs very well when the mutation rate is low, but quite poorly for large mutation rates. Moreover, the performance of *Union* degrades for an increasing number and, hence, diversity of tasks. This is particularly pronounced for high mutation rates. The three proposed methods all perform better than the two baseline methods, indicating that it pays-off to take the additional data and the relation between the tasks into account. However, among the three proposed methods there is no method that consistently performs best.

*Results on the Splice Dataset.* On the toy data set, all three MTL methods perform similarly well and clearly outperform the two baselines on all eight

<sup>4</sup> <http://fml.mpg.de/raetsch/suppl/mtl-taxonomy>



(a) Toy data: High mutation rate



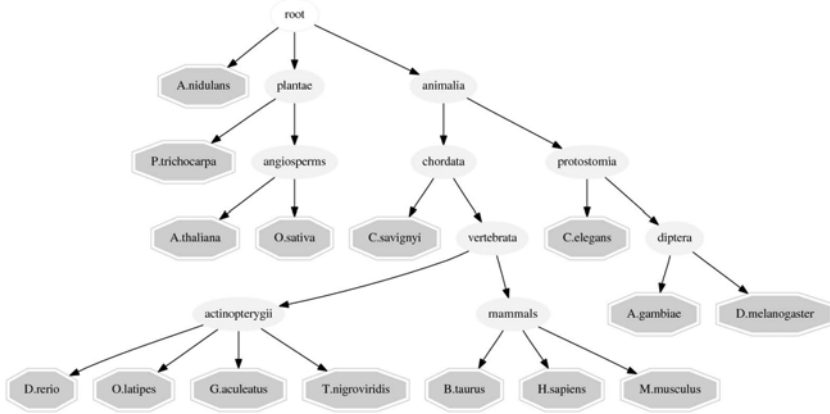
(b) Toy data: Low mutation rate

**Fig. 2.** Results for the five considered methods on the toy data sets with high and low mutation rate. Shown are the average areas under the ROC curves for different numbers of tasks that have been generated from full binary trees.

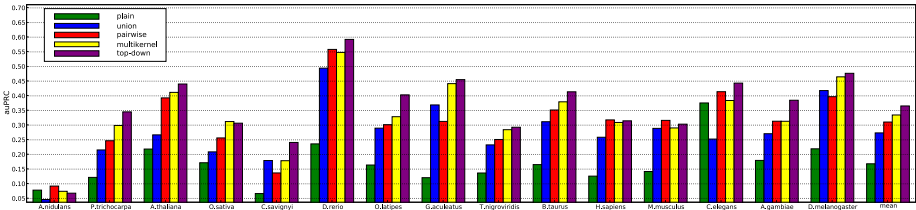
tasks. For the splice site data set, examining the average performance across all organisms (the last column in Figure 4), we also observe superior performance of the MTL methods over the baseline methods. The *Plain* method is again outperformed by all other methods, which emphasizes the importance of using data from related organisms. Comparing the hierarchical MTL methods, we observe that the *Top-Down-SVM* approach performs slightly better than the other MTL methods. We conjecture that this is due to a suboptimal choice of the similarity matrix, where tree-hop-distance was used for the pairwise and multitkernel approach.

Zooming in on the results for individual organisms, we observe the same trend: MTL methods outperform the baselines in most cases and the *Plain* method yields the worst performance. For 11 out of 15 organisms, all MTL methods perform better than the baseline. The *Top-Down* method always performs better than the baseline methods, except for *A. nidulans*, which is the only fungal organism in our set. The gain from hierarchy is more pronounced in the lower levels of the hierarchy, e.g., for *A. thaliana*, *O. sativa*, *O. latipes*, and *D. rerio*, where data from closely related organisms are used to leverage the learning process. An exception to this behaviour is seen on the mammals branch, where the performance gain from MTL methods gets smaller for *H. sapiens* and essentially diminishes for *M. musculus*. Our conjecture is that the hierarchy for this branch is not deep enough in order to represent closely related organisms. Including more similar organisms, and hence, extending the hierarchy to capture more evolutionary steps, can improve the performance gain for these organisms.

It is worthwhile to investigate the performance of the methods on *A. nidulans*, which is the child of the root node and, therefore, most distantly related to the other organisms. We observe that the *Union* method performs worst on this organism and the MTL methods perform on the same level as the *Plain* method.



**Fig. 3.** Taxonomy used to relate organisms for the splicing experiment



**Fig. 4.** Results for the splice site data sets from 15 eukaryotic genomes: Shown are auPRC performances of the five considered methods for each organism (two baseline methods: *Plain*, *Union*; three proposed methods: *Top-Down*, *Pairwise*, and *Multikernel*). We can observe that the two of the MTL methods consistently outperform the baseline methods. In 14/15 cases, the hierarchy information lead to improved prediction results.

Hence, MTL methods manage to improve the performance for closely related organisms, while causing (essentially) no performance loss on the distantly related ones.

Moreover, it is interesting to observe that *A. nidulans* and *C. elegans* are the only organisms/tasks, for which the *Union* method is considerably worse than the *Plain* method. This hints at major differences between the recognition of splice sites in these two organisms. This is less surprising for *A. nidulans* as it is for *C. elegans*. It appears worth investigating this property also for other nematode genomes to better understand this observation.

## 4 Discussion and Conclusion

We outlined two principle ways of leveraging information from related organisms, where relationship between organisms is defined with respect to a given

hierarchy. We presented three algorithms that readily deal with large scale problems such as those frequently encountered in genomic sequence analysis. We have demonstrated that our methods outperform baseline methods on synthetic data, and data from splice site prediction. On the one hand, the poor performance of *Plain* relative to the MTL methods shows that exploiting information from other tasks is in fact beneficial. On the other hand, the poor result of *Union* demonstrates that there is no single model that fits all tasks equally. Clearly, methods that carefully combine the data from different tasks according to their relatedness perform best.

We are encouraged by the good performance of the *Top-Down-SVM* method, as it provides a fast, simple and non-parametric way of exploiting hierarchical information. Inferring an accurate task-similarity matrix  $\mathbf{T}$  proves to be non-trivial, therefore one should think of additional ways of using the hierarchy to facilitate that task. Experiments on the splicing data show that a simple task-similarity matrix based on tree-hop-distance can be suboptimal, particularly for cases when edge lengths (i.e. evolutionary distance to the parent) are unequal. Our immediate future work involves experiments, where edge lengths are incorporated into the similarity matrix. For phylogenetic trees, edge lengths can, for instance, be given by evolutionary years.

From our experience with Multitask learning experiments, we conclude that certain requirements have to be fulfilled in order for MTL method to be beneficial. In particular, the problem has to be difficult enough to require considerable amounts of data. In this context, difficult means that the number of training examples for each task is relatively low, compared to the complexity of the model (e.g., we need many training examples to learn good models, when using string kernels of high degree).

Furthermore, the tasks have to be similar enough to contain mutually relevant information. If tasks are too different, and the learning problem is reasonably easy, we might be better off learning tasks independently. On the contrary, if task are too similar MTL methods will not give rise to much improvement over obtaining one global model (as shown in the toy-data experiment).

Assuming reasonable conditions in terms of problem difficulty and task similarity, we can benefit most from a hierarchy if we consider relatively many tasks. Here, we need the fine-grained information about task relations contained in a hierarchy to direct the trade-off in our learning approaches. In particular, we can benefit most compared to one global model (i.e. *Union*), if the hierarchy describes a rich structure (e.g., not a trivial one, such as all leaves attached to the root).

For all of the presented methods, we plan to provide publicly available scalable implementations based on modified versions of *SVMLight* [4] and *LibSVM* [7] as part of the Shogun Machine Learning Toolbox [12]. Lastly, we would like to emphasize that in computational biology there is a great number of problems for which corresponding data sets are available for multiple organisms. We expect that hierarchical MTL methods can indeed make a big difference for this class of problems. Therefore, the methods and implementations that we presented could be of value to a wide range of applications.

**Acknowledgments.** We would like to thank Sören Sonnenburg for help with the implementation of the presented algorithms. Also, we acknowledge Cheng Soon Ong for providing the raw splicing data sets. Furthermore, we would like to thank Gabriele Schweikert, Georg Zeller and Klaus-Robert Müller for inspiring discussions. This work was supported by the DFG grant RA1894/1-1.

## References

1. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR* 6, 1817–1853 (2005)
2. Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B., Rätsch, G.: Support vector machines and kernels for computational biology. *PLoS Comput. Biol.* 4(10), e1000173 (2008)
3. Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Wortman, J.: Learning bounds for domain adaptation. In: *Advances in Neural Information Processing Systems*, vol. 20, pp. 129–136 (2008)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001)
5. Daumé, H.: Frustratingly easy domain adaptation. In: *ACL, The Association for Computer Linguistics* (2007)
6. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* 6, 615–637 (2005)
7. Joachims, T.: *SVMLight: Support Vector Machine*. SVM-Light Support Vector Machine. University of Dortmund (1999), <http://svmlight.joachims.org/>
8. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: *Proceedings of the Pacific Symposium on Biocomputing*, pp. 564–575 (2002)
9. Rätsch, G., Sonnenburg, S.: *Accurate Splice Site Detection for Caenorhabditis elegans*. MIT Press, Cambridge (2004)
10. Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K.-R., Sommer, R., Schölkopf, B.: Improving the *C. elegans* genome annotation using machine learning. *PLoS Computational Biology* 3(2), e20 (2007)
11. Schweikert, G., Widmer, C., Schölkopf, B., Rätsch, G.: An empirical analysis of domain adaptation algorithms. In: *Advances in Neural Information Processing System, NIPS, Vancouver, B.C.*, vol. 22 (2008)
12. Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Zien, A., de Bona, F., Gehl, C., Binder, A., Franc, V.: *The shogun machine learning toolbox* (under revision). *Journal of Machine Learning Research* (2010)
13. Vapnik, V.N.: *The nature of statistical learning theory*. Springer, New York (1995)



# A Novel Abundance-Based Algorithm for Binning Metagenomic Sequences Using $l$ -Tuples

Yu-Wei Wu and Yuzhen Ye

School of Informatics and Computing, Indiana University,  
901 E. 10th Street, Bloomington, IN 47408  
{yuwuwu, yye}@indiana.edu

**Abstract.** Metagenomics is the study of microbial communities sampled directly from their natural environment, without prior culturing. Among the computational tools recently developed for metagenomic sequence analysis, binning tools attempt to classify all (or most) of the sequences in a metagenomic dataset into different bins (i.e., species), based on various DNA composition patterns (e.g., the tetramer frequencies) of various genomes. Composition-based binning methods, however, cannot be used to classify very short fragments, because of the substantial variation of DNA composition patterns within a single genome. We developed a novel approach (AbundanceBin) for metagenomics binning by utilizing the different abundances of species living in the same environment. AbundanceBin is an application of the Lander-Waterman model to metagenomics, which is based on the  $l$ -tuple content of the reads. AbundanceBin achieved accurate, unsupervised, clustering of metagenomic sequences into different bins, such that the reads classified in a bin belong to species of identical or very similar abundances in the sample. In addition, AbundanceBin gave accurate estimations of species abundances, as well as their genome sizes—two important parameters for characterizing a microbial community. We also show that AbundanceBin performed well when the sequence lengths are very short (e.g. 75 bp) or have sequencing errors.

**Keywords:** Binning, metagenomics, EM algorithm, Poisson distribution.

## 1 Introduction

Metagenomic studies have resulted in vast amounts of sequence, sampled from a variety of environments, leading to new discoveries and insights into the uncultured microbial world [1], such as the diversity of microbes in different environments [2,3], microbial (and microbe-host) interactions [4,5], and the environmental and evolutionary processes that shape these communities [6]. Current metagenomic projects are facilitated by the rapid advancement of DNA sequencing techniques. Recently developed next-generation sequencing (NGS) technologies [7] (such as Roche/454 [8] and Illumina/Solexa [9]) provide lower cost sequence, without the cloning step inherent in conventional capillary-based

methods. These NGS technologies have increased the amount of sequence data obtained in a single run by several orders of magnitude.

One of the primary goals of metagenomic projects is to characterize the organisms present in an environmental sample and identify the metabolic roles each organism plays. Many computational tools have been developed to infer species information from raw short reads directly, without the need for assembly—assembly of metagenomic sequences into genomes is extremely difficult, since the reads are often very short and are sampled from multiple genomes. We categorize the various computational tools for the estimation of taxonomic content into two basic classes, and will review them briefly below.

The first class of computational tools maps metagenomic sequences to taxa with or without using phylogeny (often referred to as the *phylotyping* of metagenomic sequences), utilizing similarity searches of the metagenomic sequences against a database of known genes/proteins. MEGAN [10] is a representative similarity-based phylotyping tool, which applies a simple lowest common ancestor algorithm to assign reads to taxa, based on BLAST results. Phylogenetic analysis of marker genes, including 16S rRNA genes [11], DNA polymerase genes [12], and the 31 marker genes defined by [13], are also applied to determining taxonomic distribution. MLTreeMap [14] and AMPHORA [15] are two phylogeny-based phylotyping tools that have been developed, using phylogenetic analysis of marker genes for taxonomic distribution estimation: MLTreeMap uses TreePuzzle [16], and AMPHORA uses PHYML [17]. CARMA [18] searches for conserved Pfam domains and protein families [19] in raw metagenomic sequences and classifies them into a higher-order taxonomy, based on the reconstruction of a phylogenetic tree for each matching Pfam family. These similarity-based and phylogeny-based phylotyping tools have a common limitation: they do not say much about the taxonomic distribution of the reads that do not match known genes/proteins, which may constitute the majority of the metagenomic sequences for some samples. A more recent approach PhymmBL [20] combines similarity search and DNA composition patterns to map metagenomic sequences to taxa, achieving an improved phylogenetic classification for short reads.

A second class of computational tools attempts to solve a related but distinct problem, the *binning* problem, which is to cluster metagenomic sequences into different bins (species). Most existing computational tools for binning utilize DNA composition. The basis of these approaches is that genome G+C content, dinucleotide frequencies, and synonymous codon usage vary among organisms, and are generally characteristic of evolutionary lineages [21]. Tools in this category include TETRA [22], MetaClust [23], CompostBin [24], TACO [25], and a genomic barcode based method [26]. All the existing DNA composition based methods achieve a reasonable performance only for long reads—at least 800 bp. TACO is able to classify genomic fragments of length 800 and 1000 bp into the phylogenetic rank of class with high accuracy (accurate classification at the order and genus level requires fragments of  $\geq 3$  kb) [25]; CompostBin was tested on simulated datasets of 1 kb reads [24]. This length limitation ( $\sim 1$  kb) will be difficult (if not impossible) to break, because of the local variation of DNA

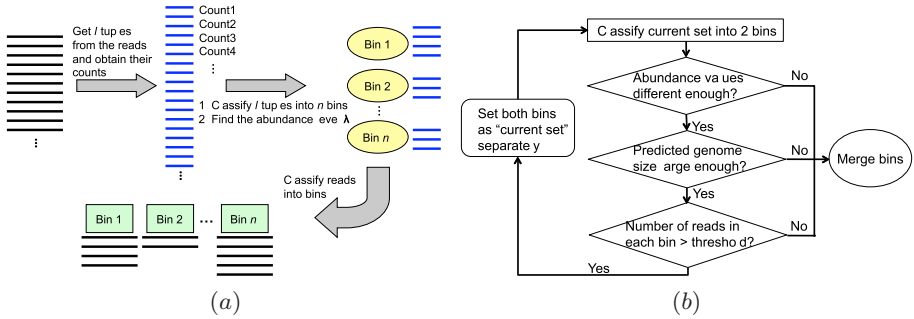
composition [21]. Foerstner et al. reported that the GC content of complex microbial communities seems to be globally and actively influenced by the environment, suggesting that it may be even harder to distinguish fragments from different species living in the same environment, based on DNA composition [27].

In addition, metagenomic sequences may be sampled from species of very different abundances (for example, the Acid Mine Drainage project [28] found two dominant species, accompanied by several other rarer species in that environment), and the difference in abundances may affect the classification results for DNA-composition based approaches. For example, a weighted PCA was adopted instead of a standard PCA in CompostBin, to reduce the dimension of composition space, considering that the within-species variance in the more abundant species might be overwhelming, compared to between-species variance [24].

Here we report a novel *binning* tool, AbundanceBin, which can be used to classify very short sequences sampled from species with different abundance levels (Fig. 1a). The fundamental assumption of our method is that reads are sampled from genomes following a Poisson distribution [29]. In the context of metagenomics, we model the sequencing reads as a mixture of Poisson distributions. We propose an Expectation-Maximization (EM) algorithm to find parameters for the Poisson distributions, which reflect the relative abundance levels of the source species. We note that a similar method was first described by Li and Waterman, for the purpose of modeling the repeat content in a conventional genome sequencing project [30], and Sharon et al. proposed a statistical framework for protein family frequency estimation from metagenomic sequences based on the Lander-Waterman model [31], given that different protein families are of different lengths. AbundanceBin assigns reads to bins using the fitted Poisson distribution. In addition, AbundanceBin gives an estimation of the genome size (or the concatenated genome size of species of the same or very similar abundances), and the coverage (which reflects the abundances of species) of each bin, all in an unsupervised manner. Since AbundanceBin is based on  $l$ -tuple content (not the composition), in principle it can be applied to classify reads that are as short as  $l$  bp. We report below first the algorithm and then tests of AbundanceBin on several synthetic metagenomic datasets and a real metagenomic dataset.

## 2 Methods

Randomized shotgun sequencing procedures result in unequal sampling of different genomes, especially when the species abundance levels differ. We seek to discover the abundance values as well as the genome sizes automatically and then bin reads accordingly. We assume that the distribution of sequenced reads follows the Lander-Waterman model [29], which calculates the coverage of each nucleotide position using a Poisson distribution. We thus view the sequencing procedure in metagenomic projects as a mixture of  $m$  Poisson distributions,  $m$  being the number of species. The goal is to find the mean values  $\lambda_1$  to  $\lambda_m$ , which are the abundance levels of the species, of these Poisson distributions.



**Fig. 1.** (a) A schematic illustration of AbundanceBin pipeline, and (b) the recursive binning approach used to automatically determine the number of bins

### 2.1 Mixed Poisson Distributions

In random shotgun sequencing of a genome, the probability that a read starting from a certain position is  $N/(G - L + 1)$ , where  $N$  is the number of reads,  $G$  is the genome size, and  $L$  is the length of reads.  $N/(G - L + 1) \approx N/G$ , given  $G \gg L$ . Assume  $x$  is a read and a  $l$ -tuple  $w$  belongs to  $x$ . The number of occurrences of  $w$  in the set of reads follows a Poisson distribution with parameter  $\lambda = N(L - l + 1)/(G - L + 1) \approx NL/G$  in a random sampling process with read length  $L$ .

Similarly, for a metagenomic dataset, the number of occurrences of  $w$  in the set of reads also follows a Poisson distribution with parameter  $\lambda = N(L - l + 1)/(G - L + 1) \approx NL/G$ , but  $G$  in this case is the total length of the genomic sequences contained in the metagenomic dataset. In metagenomic datasets, the reads are from species with different abundances. If the abundance of a species  $i$  is  $n$ , the total number of occurrences of  $w$  in the whole set of reads coming from this species should follow a Poisson distribution with parameter  $\lambda_i = n\lambda$ , due to the additivity of Poisson distribution. Now the problem of finding the relative abundance levels of different species is transformed to the modeling of mixed Poisson distributions.

### 2.2 Binning Algorithm

Given a set of metagenomic sequences, the algorithm starts by counting  $l$ -tuples in all reads (Fig. 1a). Then we use an Expectation-Maximization (EM) algorithm to approximate the species abundance level and the genome size of each species, which consists of 4 steps, as follows.

1. Initialize the total number of species  $S$ , their genome size  $l_i$ , and abundance level  $\lambda_i$  for  $i = 1, 2, \dots, S$ .

2. Calculate the probability that the  $l$ -tuple  $w_j$  ( $j = 1, 2, \dots, W$ ;  $W$  is the total number of possible  $l$ -tuples) coming from  $i$ th species given its count  $n(w_j)$  (see Appendix for details).

$$P(w_j \in s_i | n(w_j)) = \frac{l_i}{\sum_{m=1}^S l_m \left(\frac{\lambda_m}{\lambda_i}\right)^{n(w_j)} e^{(\lambda_i - \lambda_m)}} \tag{1}$$

3. Calculate the new values for each  $l_i$  and  $\lambda_i$ .

$$l_i = \sum_{j=1}^W P(w_j \in s_i | n(w_j)) \tag{2}$$

$$\lambda_i = \frac{\sum_{j=1}^W n(w_j) P(w_j \in s_i | n(w_j))}{l_i} \tag{3}$$

4. Iterate step 2 and 3 until the parameters converge or the number of runs exceeds a maximum number of runs. The convergence of parameters is defined as

$$\forall \lambda_i \left\{ \left| \frac{\lambda_i^{t+1}}{\lambda_i^t} \right| < 10^{-5} \right\} \text{ and } \forall l_i \left\{ \left| \frac{l_i^{t+1}}{l_i^t} \right| < 10^{-5} \right\} \tag{4}$$

where  $\lambda_i^{(t)}$  and  $l_i^{(t)}$  represent the abundance level and genome length at iteration  $t$ , respectively. The maximum number of runs is set to 100 (which is sufficient for the convergence of the EM algorithm for all the cases we have tested).

Once the EM algorithm converges, we can estimate the probability of a read assigned to a bin, based on its  $l$ -tuples binning results as,

$$P(r_k \in s_i) = \frac{\prod_{w_j \in r_k} P(w_j \in s_i | n(w_j))}{\sum_{s_i \in S} \left( \prod_{w_j \in r_k} P(w_j \in s_i | n(w_j)) \right)} \tag{5}$$

where  $r_k$  is a given read,  $w_j$  is the  $l$ -tuples that belong to  $r_k$ , and  $s_i$  is any bin. A read will be assigned to the bin with the highest probability among all bins. A read remains unassigned if 90% of its  $l$ -tuples are excluded (counts too low or too high, see section 2.3), or if the highest probability is  $< 50\%$ .

### 2.3 Lower- and Upper-Limit for $l$ -Tuple Counts

We set a lower- and upper-limit for  $l$ -tuple counts, as additional parameters, when we approximate  $\lambda_i$  and  $l_i$ . The lower-limit is introduced to deal with sequencing errors, and the upper-limit is introduced to handle  $l$ -tuples with extremely high counts, such as those from vector sequences or repeats of high copy numbers. Let the lower-limit be  $B_{lower}$  and the upper-limit be  $B_{upper}$ . Then the formula for calculating  $\lambda_i$  and  $l_i$  is modified to

$$l_i = \sum_{j=1}^W P(w_j \in s_i | n(w_j)), \forall n(w_j) > B_{lower} \wedge n(w_j) < B_{upper} \tag{6}$$

$$\lambda_i = \frac{\sum_{j=1}^W n(w_j)P(w_j \in s_i | n(w_j))}{l_i}, \forall n(w_j) > B_{lower} \wedge n(w_j) < B_{upper} \quad (7)$$

## 2.4 Automatic Determination of the Total Number of Bins by a Recursive Binning Approach

In the EM algorithm, we need to provide the number of bins as an input, in order to determine the parameters of the mixed Poisson distributions. However, this number is often unknown, as for most metagenomic projects. We implemented a recursive binning approach to determine the total number of bins automatically. The recursive binning approach works by separating the dataset into two bins and proceeds by further splitting bins (as shown in Fig. 1b)—it is a top-down approach. The recursive binning approach was motivated by the observation that reads from genomes with higher abundances are better classified than those with lower abundance. The recursive procedure continues if 1) the predicted abundance values of two bins differ significantly, i.e.,  $|\lambda_i - \lambda_j| / \min(\lambda_i, \lambda_j) \geq 1/2$ ; 2) the predicted genome sizes are larger than a certain threshold (currently set to 400,000, considering that the smallest genomes of living organisms yet found are about 500,000 bp—*Nanoarchaeum equitans* has a genome of 490,885 bp, and *Mycoplasma genitalium* has a genome of 580,073 bp); and 3) the number of reads associated with each bin is larger than a certain threshold proportion (3%) of the total number of reads classified in the parent bin.

## 2.5 Performance Evaluation

We defined the classification error rate as the number of misclassified reads divided by the total number of reads. Chatterji et al [24] used a normalized error rate—the arithmetic average of the classification error rates for all the bins—to evaluate their binning approach CompostBin. We consider that the standard error rate, instead of normalized error rate, serves better for the performance evaluation of AbundanceBin, since AbundanceBin takes advantage of different species abundances. For comparison, we also provide the normalized classification error rates.

## 2.6 Metagenomic Datasets

We used MetaSim [32] to generate synthetic metagenomic datasets with reads sampled from species of various abundances. MetaSim takes as input a set of known genome sequences and an abundance profile, which determines the relative abundance of each genome sequence in a simulated dataset. The “Exact” profile defined by MetaSim is used to generate reads without sequencing errors, and “454” profile for reads generated with a 454 error model. The number of reads as well as the mean and variance of read lengths are adjusted accordingly: for average 400 bp, the mean value is set to 400 and the variance is set to 50;

and for 75 bp, mean and variance are set to 75 and 5. All other settings are kept as default. The genomes we used for generating synthetic metagenomic datasets, and the AMD metagenomic sequences and its scaffolds were downloaded from NCBI.

### 3 Results

We tested AbundanceBin on various synthetic metagenomic datasets with short and very short sequence lengths (75–400 bp), and the results show that AbundanceBin gives accurate classification of reads to different bins, and accurate estimation of the abundances—as well as the genome sizes—in each bin. We note that since these parameters are usually unknown in real metagenomic datasets, we focused on synthetic datasets for benchmarking. We also applied AbundanceBin to the actual AMD dataset and revealed a relatively clear picture of the complexity of the microbial community in that environment, consistent with the analysis reported in [28].

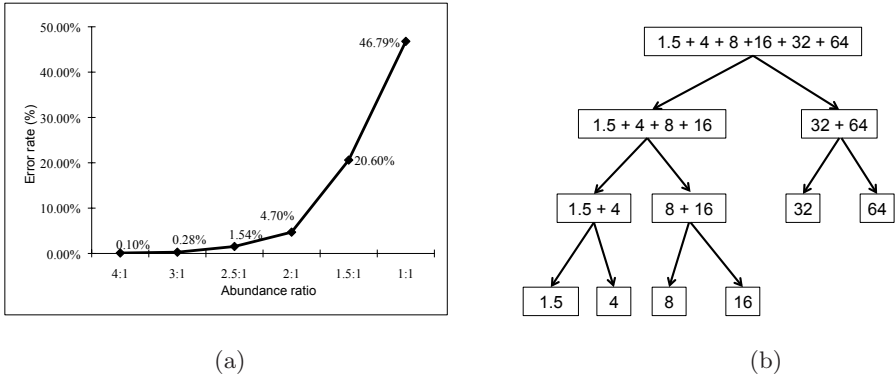
#### 3.1 Tests of Abundance Differences and the Length of $l$ -Tuples

We did a series of experiments to test the abundance ranges of species required for accurate binning of reads. Fig. 2a shows the binning results for simulated short reads sampled from two genomes (*Mycoplasma genitalium* G37 and *Buchnera aphidicola* str. BP) at abundance ratios, 4:1, 3:1, 2.5:1, 2:1, 1.5:1, and 1:1 (with 50,000 simulated reads of  $\sim 400$  bases for each setting). The classification error rate is low if the abundance ratio is  $\geq 2.0$  (0.1% and 4.7% for ratio 4:1 and 2:1, respectively), but rises dramatically when the abundance ratio drops to 1.5:1 (the error rate is 20.6% for abundance ratio 1.5:1). We conclude that the abundance ratio needs to reach at least 2:1 for a good classification by AbundanceBin.

We also tested different lengths of  $l$ -tuples, and the results show that when  $l$  drops to 16, the binning performance dropped significantly for cases with two genomes. The performance improved slightly when  $l$  increases to 20 for cases with more than 3 genomes. Considering the performance on the tested cases, we chose to use  $l = 20$  for the following experiments.

#### 3.2 AbundanceBin Achieves Accurate Binning, Estimation of Species Abundance, and Genome Size

The binning results on several simulated datasets of short reads are summarized in Table 1. AbundanceBin achieved both accurate estimation of species abundances, and accurate assignment of reads to bins of different abundances. The classification error rates are 0.10% and 0.64% for the classification of reads of length 400 bp and 75 bp, respectively, sampled from two genomes (cases A and C in Table 1). The error rates for the classification of reads sampled from more



**Fig. 2.** (a) The classification error rates for classifying reads sampled from two genomes versus their abundance differences, and (b) the recursive binning of a read dataset into 6 bins of different abundances (each box represents a bin with the numbers indicating the abundance of the reads classified to that bin; e.g., the bin on the top has all the reads, which will be divided into two bins, one with reads of abundances 1.5, 4, 8 and 64, and the other bin with reads of abundances 32 and 64)

genomes are slightly higher than for two-genome scenarios (e.g., the classification error rates for two synthetic metagenomic datasets with reads of length 400 bp and 75 bp, sampled from 3 genomes, are 3.10% and 6.18%, respectively, as shown in Table 1). For the classification of reads sampled from more than two genomes, most of the errors occur in the least abundant bin. But AbundanceBin was still able to classify the reads from species of higher abundance correctly for all the tested synthetic metagenomic datasets, including one with reads sampled from 6 different genomes (see Table 2).

We emphasize here that AbundanceBin can bin reads as short as 75 bases with reasonable classification error rates, as shown in Table 1. As we discussed in the Introduction, binning of very short reads, such as 75 bases, is extremely difficult and cannot be achieved by any of the existing composition based binning approaches, due to the substantial variation in DNA composition within a single genome. AbundanceBin will also give an estimation of the genome size for each bin. As shown in Table 1, for most of the tested cases, the estimated genome sizes are very close to the real ones. We note that AbundanceBin will classify reads from different species of similar abundances into a single bin. In this case, the predicted genome size for that bin is actually the sum of the genome sizes of the species classified into that bin.

AbundanceBin also worked well on binning closely related species (closely related species often have similar genomes, and therefore it is often very difficult to separate reads sampled from closely related species). For the synthetic metagenomic datasets we tested, most reads from species that differ at only the species level can still be classified into correct bins with very low error rates. For examples, for two datasets, AbundanceBin resulted in binning with error rates



of 0.96% and 0.68% for the dataset simulated from the genomes of *Corynebacterium efficiens* YS-314 and *Corynebacterium glutamicum* ATCC 13032, and the dataset simulated from the genomes of *Helicobacter hepaticus* ATCC 51449 and *Helicobacter pylori* 26695 (both sets of genomes only differ at the species level), respectively. These results demonstrate the ability of our algorithm to separate short reads from closely related species, even if the species are of the same genus. (Note that AbundanceBin cannot separate reads from different strains of the same species into different bins.)

### 3.3 AbundanceBin Can Handle Sequencing Errors

As mentioned in Methods, AbundanceBin can be configured to ignore  $l$ -tuples that only appear once to deal with sequencing errors, considering that those  $l$ -tuples are likely to be contributed by reads with sequencing errors and that the chance of having reads with sequencing errors at the same position will be extremely low. This may exclude some genuine  $l$ -tuples, but our tests reveal that AbundanceBin achieved even better classification if all  $l$ -tuples of count 1 are discarded (data not shown). AbundanceBin achieved slightly worse classification of reads when reads contain sequencing errors, as compared to the classification of simulated reads without sequencing errors (see cases E and F in Table 1). This is expected, given that many spurious  $l$ -tuples are generated with a 454 sequencing error model. For example, 12,901,691 20-tuples can be found in a dataset of simulated reads from two genomes with sequencing errors (case E in Table 1), 5 times more than the case without error models (2,370,720).

### 3.4 AbundanceBin Doesn't Require Prior Knowledge of the Total Number of Bins

Table 2 compares the performance of AbundanceBin using the recursive binning approach on several synthetic metagenomic datasets to that of AbundanceBin given the total number of bins. Overall the performances of the recursive binning approach are comparable to the cases with predefined bin numbers. Fig. 2b depicts the recursive binning results of the classification of one of the synthetic metagenomic datasets (which has reads sampled from 6 genomes) into 6 bins of different abundances (with classification error rate = 3.73%), starting with a bin that includes all the reads and ending with 6 bins each having reads correctly assigned to them. It is interesting that the recursive binning approach achieved even better performance for some cases. A simple explanation to this is that the recursive binning strategy may create bigger abundance differences, especially at the beginning of the binning process, and AbundanceBin works better at separating reads from species with greater abundance differences (see Fig. 2a). We note again that the high abundant bins are classified relatively well. The majority of errors occur in low abundant bins.

**Table 1.** Tests of AbundanceBin on synthetic metagenomic datasets (A-D without sequencing errors, and E-F with sequencing errors <sup>a</sup>)

ID	Spe <sup>b</sup>	Len <sup>c</sup>	Total reads	Bin	Abundance		Genome size		Error rate(%)
					Real	Predicted	Real	Predicted	
A	2	400 bp	50,000	1	27.23	26.27	580,076	570,859	0.10 (0.20 <sup>d</sup> )
				2	6.83	6.49	615,980	614,605	
B	3	400 bp	50,000	1	24.64	23.78	580,076	568,549	3.10 (6.64)
				2	6.13	6.02	615,980	517,110	
				3	1.8	2.39	1,072,950	941,425	
C	2	75 bp	200,000	1	20.47	15.66	580,076	562,584	0.64 (1.07)
				2	5.08	3.92	615,980	608,401	
D	3	75 bp	200,000	1	27.6	20.93	580,076	565,859	6.18 (11.74)
				2	6.93	5.99	615,980	368,836	
				3	2.07	2.43	1,072,950	1,100,309	
E	2	297 bp	50,000	1	20.21	11.63	580,076	521,168	1.12 (0.99)
				2	5.07	3.01	615,980	945,435	
F	3	297 bp	150,000	1	55.48	30.58	580,076	559,395	8.20 (11.41)
				2	13.98	9.6	615,980	341,290	
				3	3.50	2.72	1,072,950	3,064,199	

<sup>a</sup>: The average sequencing error rate introduced is 3%, higher than the error rate of recent 454 machines (e.g., the accuracy rate reported in [33] is 99.5%). A 3% sequencing error can reduce the  $l$ -tuple counts by about half (i.e., about  $1 - 0.97^{20} = 0.46$  of expected 20-mers without sequencing errors), which makes accurate estimation of abundance and genome size difficult. <sup>b</sup>: The number of species used in simulating each metagenomic dataset. The genomes used in these tests are *Mycoplasma genitalium* G37, *Buchnera aphidicola* str. BP, and *Chlamydia muridarum* Nigg. The first two genomes are used for the 2 species cases. <sup>c</sup>: The average length of the simulated reads. <sup>d</sup>: Normalized error rates (see Methods for details).

**Table 2.** Comparison of binning performance using the recursive binning approach (“Recursive”) versus the binning when the total number of bins is given (“Predefined”)

Test cases	Error rate (normalized error rate)	
	Predefined	Recursive
3 genomes (no error model; 400 bp)	3.10% (6.64%)	3.24% (7.47%)
3 genomes (no error model; 75 bp)	6.18% (11.74%)	4.84% (9.31%)
3 genomes (454 error model; 297 bp)	8.21% (11.41%)	2.29% (4.21%)
4 genomes (no error model; 400bp)	1.12% (5.16%)	2.96% (6.96%)
6 genomes (no error model; 400bp)	2.50% (9.23%)	3.73% (13.07%)

### 3.5 Binning of Acid Mine Drainage (AMD) Datasets

The AMD microbial community was reported to consist of two species of high abundance and three other less abundant species [28]. With the difference of two abundance levels in this environment, we expect that the algorithm could classify the AMD dataset into two bins. We applied AbundanceBin to a simulated AMD

dataset (so that we have correct answers for comparison) and the real AMD dataset from [28].

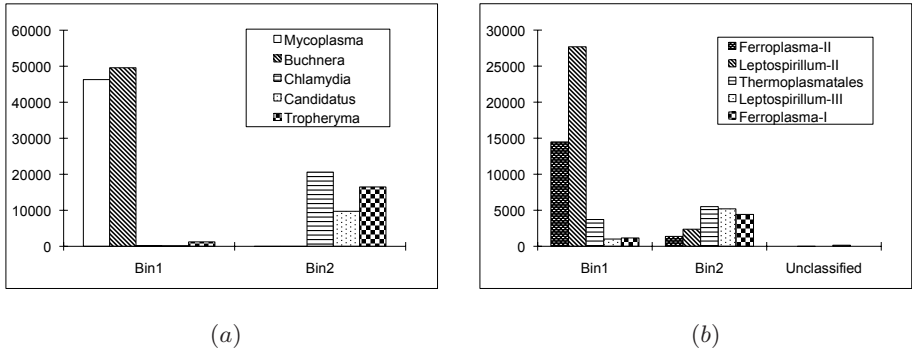
The synthetic AMD dataset contains 150,000 reads from five genomes, with abundances 4:4:1:1:1. Our recursive binning approach automatically classified the reads into two bins with an error rate of 1.03% (see Fig. 3a). (Note here that each bin has reads sampled from multiple species. We consider that a read is classified correctly if it is classified into the bin of the correct abundance.) The binning accuracy dropped only slightly (with an error rate of 2.25%) for the synthetic AMD dataset when sequencing errors were introduced.

We also applied AbundanceBin to reads from the actual AMD dataset (downloaded from NCBI trace archive; 13696\_environmental\_sequence.007). AbundanceBin successfully classified these reads into exactly two bins (one of high abundance and one of low abundance) using the recursive binning approach (see Fig. 3b). Note the reads in this dataset have vector sequences, which resulted in a very small number of  $l$ -tuples of extremely high abundance (the highest count is 50,720)—this phenomena has been utilized for vector sequence removal, as described in [34]. Two approaches were employed to avoid the influences of the vector sequences: 1) we used the Figaro software package [34] to trim the vector sequences, and 2) we set an upper-limit for the count of all  $l$ -tuples, ignoring  $l$ -tuples with counts larger than the upper-limit (200 by default). We also downloaded the sequences of 5 scaffolds of the 5 partial genomes assembled from the AMD dataset, so that we can estimate the classification accuracy of AbundanceBin. The classification error rate of the AMD sequences is  $\sim 14.38\%$ . Note this error rate only gives us a rough estimation of the classification accuracy, since only 58% of the AMD reads can be mapped back to the assembled scaffolds based on similarity searches by BLAST—we mapped a read to a scaffold if the read matches the scaffold with BLAST E-value  $\leq 1e-50$ , sequence similarity  $\geq 95\%$ , and a matched length of  $\geq 70\%$  of the read length. We emphasize that AbundanceBin achieved a much better classification (with an error rate of 1.03%) for the simulated AMD reads, for which we have correct answers to compare with.

## 4 Discussion

We have shown that our abundance based algorithm for binning has the ability to classify short reads from species with different abundances. Our approach has two unique features. First, our method is “unsupervised” (i.e., it doesn’t require any prior knowledge for the binning). Second, our method is especially suitable for short reads, as long as the length of reads exceeds the length of the  $l$ -tuple (currently 20). AbundanceBin can in principle be applied to any metagenomic sequences acquired by current NGS, without human interpretation.

We implemented a simple strategy—excluding  $l$ -tuples that are counted only once from the abundance estimation—to handle sequencing errors, and tests have showed that AbundanceBin achieved better classification if all  $l$ -tuples of single count are discarded. One potential problem of discarding  $l$ -tuples of low counts is that some genuine  $l$ -tuples will be discarded as well, which results



**Fig. 3.** The binning results for a simulated (a), and the actual (b) AMD datasets. The histogram shows the total number of reads from different genomes classified to each bin.

in a lower abundance estimation and a worse prediction of genome sizes, especially for the species with low abundance, as shown in Table 1. But we argue that AbundanceBin can still capture the relative abundances of different bins correctly, which is more important than the absolute values. Another potential problem is that reads from low abundant genomes may not be classified when sequencing errors are introduced in the reads. For example, the number of unclassified reads in a two-genome case (metagenomic dataset E in Table 1) is 12, and 389 in a three-genome case (metagenomic dataset F in Table 1). All unclassified reads in both cases belong to the least abundant species, indicating that the abundance values greatly affect the predicted results, especially when sequencing errors are present. We expect that both problems will become less problematic as sequencing coverage is increased, which is possible with massive throughput NGS techniques. As for the abundance ratio required for successful classification, we find that the ratio should be at least 2:1 to obtain an acceptable result. The required ratio, of course, is also affected by several other factors, such as the actual abundance level, the average length of reads, and the sequencing error rate. Our tests intentionally use well-classified datasets to allow us to follow changes in classification error resulting from abundance differences, but other factors besides the abundance ratio must also be considered.

AbundanceBin runs fast, and all the tests shown in the paper were completed within an hour (using single CPU on Intel(R) Xeon(R)@2.00GHz) with very moderate memory usage. For example, binning of the synthetic metagenomic dataset A (see Table 1) requires 100MB memory, and dataset B 150MB. However, AbundanceBin may require large memory when working with very large datasets of short reads.

We expect that AbundanceBin will have three important applications. First it can be used for binning metagenomic sequences, as well as estimating species abundances and genome sizes. Second, it can be combined with other binning approaches. Note that AbundanceBin is not designed to separate reads from

species of very similar abundances; we will develop an integrated method that combines AbundanceBin and other binning methods that use, for example, DNA composition. We expect that such an integrated method will achieve better classification performance by incorporating orthogonal information (abundance and composition, for example). Finally, we expect that applying AbundanceBin to separate reads into bins of different abundances (coverages), prior to the assembly of metagenomic sequences, will improve the quality of genome assembly.

## Acknowledgements

This research was supported by NIH grant 1R01HG004908-02 and NSF CAREER award DBI-084568. We thank Dr. Haixu Tang for helpful discussions, and Dr. Thomas G. Doak for reading the manuscript.

## References

- Galperin, M.: Metagenomics: from acid mine to shining sea. *Environ. Microbiol.* 6, 543–545 (2004)
- Tringe, S., von Mering, C., Kobayashi, A., et al.: Comparative metagenomics of microbial communities. *Science* 308(5721), 554–557 (2005)
- Dinsdale, E., Pantos, O., Smriga, S., et al.: Microbial ecology of four coral atolls in the northern line islands. *PLoS ONE* 3(2), e158 (2008)
- Turnbaugh, P.J., Ley, R.E., Mahowald, M.A., et al.: An obesity-associated gut microbiome with increased capacity for energy harvest. *Nature* 444(7122), 1027–1131 (2006)
- Turnbaugh, P.J., Hamady, M., Yatsunencko, T., et al.: A core gut microbiome in obese and lean twins. *Nature* 457(7228), 480–484 (2009)
- Dinsdale, E.A., Edwards, R.A., Hall, D., et al.: Functional metagenomic profiling of nine biomes. *Nature* 452(7187), 629–632 (2008)
- Hutchison Jr., C.A.: DNA sequencing: bench to bedside and beyond. *Nucleic Acids Res.* 35(18), 6227–6237 (2007)
- Margulies, M., Egholm, M., Altman, W.E., et al.: Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437(7057), 376–380 (2005)
- Bentley, D.R.: Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.* 16(6), 545–552 (2006)
- Huson, D.H., Auch, A.F., Qi, J., et al.: MEGAN analysis of metagenomic data. *Genome Res.* 17(3), 377–386 (2007)
- Chakravorty, S., Helb, D., Burday, M., et al.: A detailed analysis of 16s ribosomal RNA gene segments for the diagnosis of pathogenic bacteria. *J. Microbiol. Methods* 69(2), 330–339 (2007)
- Monier, A., Claverie, J.M., Ogata, H.: Taxonomic distribution of large DNA viruses in the sea. *Genome Biol.* 9(7), R106 (2008)
- Ciccarelli, F.D., Doerks, T., von Mering, C., et al.: Toward automatic reconstruction of a highly resolved tree of life. *Science* 311(5765), 1283–1287 (2006)
- von Mering, C., Hugenholtz, P., Raes, J., et al.: Quantitative phylogenetic assessment of microbial communities in diverse environments. *Science* 315(5815), 1126–1130 (2007)

15. Wu, M., Eisen, J.A.: A simple, fast, and accurate method of phylogenomic inference. *Genome Biol.* 9(10), 151 (2008)
16. Schmidt, H.A., Strimmer, K., Vingron, M., et al.: TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics* 18(3), 502–504 (2002)
17. Guindon, S., Gascuel, O.: A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.* 52(5), 696–704 (2003)
18. Krause, L., Diaz, N.N., Goesmann, A., et al.: Phylogenetic classification of short environmental DNA fragments. *Nucleic Acids Res.* 36(7), 2230–2239 (2008)
19. Finn, R.D., Mistry, J., Schuster-Bockler, B., et al.: Pfam: clans, web tools and services. *Nucleic Acids Res.* 34(Database issue), D247–D251 (2006)
20. Brady, A., Salzberg, S.L.: Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nat. Methods* 6(9), 673–676 (2009)
21. Bentley, S.D., Parkhill, J.: Comparative genomic structure of prokaryotes. *Annu. Rev. Genet.* 38, 771–792 (2004)
22. Teeling, H., Waldmann, J., Lombardot, T., et al.: TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. *BMC Bioinformatics* 5, 163 (2004)
23. Woyke, T., Teeling, H., Ivanova, N.N., et al.: Symbiosis insights through metagenomic analysis of a microbial consortium. *Nature* 443(7114), 950–955 (2006)
24. Chatterji, S., Yamazaki, I., Bai, Z., et al.: CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 17–28. Springer, Heidelberg (2008)
25. Diaz, N.N., Krause, L., Goesmann, A., et al.: TACOA: taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. *BMC Bioinformatics* 10, 56 (2009)
26. Zhou, F., Olan, V., Xu, Y.: Barcodes for genomes and applications. *BMC Bioinformatics* 9, 546 (2008)
27. Foerstner, K.U., von Mering, C., Hooper, S.D., et al.: Environments shape the nucleotide composition of genomes. *EMBO Rep.* 6(12), 1208–1213 (2005)
28. Tyson, G.W., Chapman, J., Hugenholtz, P., et al.: Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428(6978), 37–43 (2004)
29. Lander, E.S., Waterman, M.S.: Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* 2(3), 231–239 (1988)
30. Li, X., Waterman, M.S.: Estimating the repeat structure and length of DNA sequences using  $l$ -tuples. *Genome Res.* 13(8), 1916–1922 (2003)
31. Sharon, I., Pati, A., Markowitz, V.M., et al.: A statistical framework for the functional analysis of metagenomes. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 496–511. Springer, Heidelberg (2009)
32. Richter, D.C., Ott, F., Auch, A.F., et al.: MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS ONE* 3(10), e3373 (2008)
33. Huse, S.M., Huber, J.A., Morrison, H.G., Sogin, M.L., Welch, D.M., et al.: Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol.* 8(7), 143 (2007)
34. White, J.R., Roberts, M., Yorke, J.A., et al.: Figaro: a novel statistical method for vector sequence removal. *Bioinformatics* 24(4), 462–467 (2008)

## Appendix

Equation (1) is used to calculate the probability that  $l$ -tuple  $w_j$  ( $j=1,2,\dots,W$ ;  $W$  is the total number of possible  $l$ -tuples) coming from the  $i$ th species given its count  $n(w_j)$ . It is computed by applying Bayes' rule as follows.

$$\Pr(w_j \in s_i | n(w_j)) = \frac{\Pr(n(w_j) | w_j \in s_i) \Pr(w_j \in s_i)}{\Pr(n(w_j))} \tag{8}$$

$$= \frac{\Pr(n(w_j) | w_j \in s_i) \Pr(w_j \in s_i)}{\sum_{m=1}^S \Pr(n(w_j) \in s_m | w_j \in s_m) \Pr(w_j \in s_m)} \tag{9}$$

$$= \frac{\Pr(n(w_j) | w_j \in s_i) \cdot \frac{l_i}{G}}{\sum_{m=1}^S \Pr(n(w_j) \in s_m | w_j \in s_m) \cdot \frac{l_m}{G}} \tag{10}$$

$$= \frac{\frac{\lambda_i^{n(w_j)} e^{-\lambda_i}}{n(w_j)!} \cdot l_i}{\sum_{m=1}^S \left( \frac{\lambda_m^{n(w_j)} e^{-\lambda_m} \cdot l_m}{n(w_j)!} \right)} \tag{11}$$

$$= \frac{l_i}{\sum_{m=1}^S \left( \left( \frac{\lambda_m}{\lambda_i} \right)^{n(w_j)} \cdot e^{\lambda_i - \lambda_m} \cdot l_m \right)} \tag{12}$$

where  $\Pr(w_j \in s_i) = \frac{l_i}{G}$  is the prior probability that word  $j$  is from species  $i$ , and  $G$  is the total length of genomic sequences contained in the metagenomic dataset. Equation (12) is the result of applying the probability mass function of Poisson distribution into the probability function of equation (10).

# A Markov Random Field Framework for Protein Side-Chain Resonance Assignment\*

Jiayang Zeng<sup>1</sup>, Pei Zhou<sup>2</sup>, and Bruce Randall Donald<sup>1,2,\*\*</sup>

<sup>1</sup> Department of Computer Science, Duke University, Durham, NC 27708, USA

<sup>2</sup> Department of Biochemistry, Duke University Medical Center, Durham, NC 27708, USA

Tel.: 919-660-6583; Fax: 919-660-6519

brd+recomb1.0@cs.duke.edu

**Abstract.** Nuclear magnetic resonance (NMR) spectroscopy plays a critical role in structural genomics, and serves as a primary tool for determining protein structures, dynamics and interactions in physiologically-relevant solution conditions. The current speed of protein structure determination via NMR is limited by the lengthy time required in resonance assignment, which maps spectral peaks to specific atoms and residues in the primary sequence. Although numerous algorithms have been developed to address the *backbone* resonance assignment problem [68, 2, 10, 37, 14, 64, 11, 31, 60], little work has been done to automate *side-chain* resonance assignment [43, 48, 5]. Most previous attempts in assigning side-chain resonances depend on a set of NMR experiments that record through-bond interactions with side-chain protons for each residue. Unfortunately, these NMR experiments have low sensitivity and limited performance on large proteins, which makes it difficult to obtain enough side-chain resonance assignments. On the other hand, it is essential to obtain almost all of the side-chain resonance assignments as a prerequisite for high-resolution structure determination. To overcome this deficiency, we present a novel side-chain resonance assignment algorithm based on alternative NMR experiments measuring through-space interactions between protons in the protein, which also provide crucial distance restraints and are normally required in high-resolution structure determination. We cast the side-chain resonance assignment problem into a Markov Random Field (MRF) framework, and extend and apply combinatorial protein design algorithms to compute the optimal solution that best interprets the NMR data. Our MRF framework captures the contact map information of the protein derived from NMR spectra, and exploits the structural information available from the backbone conformations determined by orientational restraints and a set of discretized side-chain conformations (i.e., rotamers). A Hausdorff-based computation is employed in the scoring function to evaluate the probability of side-chain resonance assignments to generate the observed NMR spectra. The complexity of the assignment problem is first reduced by using a *dead-end elimination* (DEE) algorithm, which prunes side-chain resonance assignments that are *provably* not part of the optimal solution. Then an A\* search algorithm is used to find a set of optimal side-chain resonance assignments that best fit the NMR data. We have tested our algorithm

---

\* This work is supported by the following grants from National Institutes of Health: R01 GM-65982 to B.R.D. and R01 GM-079376 to P.Z.

\*\* Corresponding author.



on NMR data for five proteins, including the FF Domain 2 of human transcription elongation factor CA150 (FF2), the B1 domain of Protein G (GB1), human ubiquitin, the ubiquitin-binding zinc finger domain of the human Y-family DNA polymerase Eta (pol  $\eta$  UBZ), and the human Set2-Rpb1 interacting domain (hSRI). Our algorithm assigns resonances for more than 90% of the protons in the proteins, and achieves about 80% correct side-chain resonance assignments. The final structures computed using distance restraints resulting from the set of assigned side-chain resonances have backbone RMSD 0.5 – 1.4 Å and all-heavy-atom RMSD 1.0 – 2.2 Å from the reference structures that were determined by X-ray crystallography or traditional NMR approaches. These results demonstrate that our algorithm can be successfully applied to automate side-chain resonance assignment and high-quality protein structure determination. Since our algorithm does not require any specific NMR experiments for measuring the through-bond interactions with side-chain protons, it can save a significant amount of both experimental cost and spectrometer time, and hence accelerate the NMR structure determination process.

## 1 Introduction

The knowledge of the 3D structures of proteins plays an important role in understanding protein functions and discovering new drugs. Although high-throughput DNA sequencing technologies have been able to identify nearly the complete sequence of the human genome, studies of the 3D structures of proteins on a genome-wide scale (i.e., structural proteomics) are still limited by current slow speed of protein structure determination. X-ray crystallography and nuclear magnetic resonance (NMR<sup>1</sup>) are two primary experimental methods for high-resolution protein structure determination. Unfortunately, structure determination by either method is laborious and time-consuming. In X-ray crystallography, growing a good quality crystal is in general a difficult task. NMR structure determination does not require crystals, thus it can be used to determine protein structures in the physiologically-relevant solution state, and has become a premier tool for studying protein dynamics. However, current NMR structure determination is still limited by the lengthy time required to process and analyze the experimental data. The development of automated and efficient procedures for analyzing NMR data and acquiring experimental restraints will thereby speed up protein structure determination and advance structural proteomics research. In practice, *side-chain resonance assignments* (the focus of this paper) are required for both side-chain dynamics studies and high-resolution structure determination.

---

<sup>1</sup> Abbreviations used: NMR, nuclear magnetic resonance; ppm, parts per million; RMSD, root mean square deviation; HSQC, heteronuclear single quantum coherence spectroscopy; NOE, nuclear Overhauser effect; NOESY, nuclear Overhauser and exchange spectroscopy; TOCSY, total correlation spectroscopy; TROSY, transverse relaxation-optimized spectroscopy; RDC, residual dipolar coupling; PDB, Protein Data Bank; BMRB, Biological Magnetic Resonance Bank; pol  $\eta$  UBZ, ubiquitin-binding zinc finger domain of the human Y-family DNA polymerase Eta; hSRI, human Set2-Rpb1 interacting domain; FF2, FF Domain 2 of human transcription elongation factor CA150; GB1, B1 domain of Protein G; CH, C $^{\alpha}$ -H $^{\alpha}$ ; SSE, secondary structure element; C', carbonyl carbon; MRF, Markov Random Field; DEE, dead-end elimination; GMEC, global minimum energy conformation.

In NMR terminology, each atom in the known primary sequence of a target protein is represented by a unique *chemical shift* (or *resonance*) in NMR spectra, that is, chemical shift serves as a scalar “ID” for an atom in the primary sequence. The magnetic interactions captured by an NMR spectrum can be described as a graph, in which each node represents the resonance of an atom in the primary sequence, and each edge represents a possible atomic interaction either through bond or through space. We call such a graph the *NMR interaction graph* [2]. For example, in an NMR interaction graph derived from a *heteronuclear single quantum coherence spectroscopy* (HSQC) spectrum, each edge represents an amide bond (i.e.,  $\text{H}^{\text{N}}-\text{N}$ ) interaction, while in an NMR interaction graph derived from a *nuclear Overhauser and exchange spectroscopy* (NOESY) spectrum, each edge represents a through-space interaction between a pair of protons closer than 6 Å, measured via the *nuclear Overhauser effect* (NOE).

In general, NMR structure determination is accomplished through the following procedure. The first step is to identify the correspondence between chemical shifts (i.e., nodes in the NMR interaction graph) and atoms in the primary sequence. Such a process is called *resonance assignment*, which is a crucial step in NMR data analysis and structure calculation. The resonance assignment can be classified into two categories: *backbone resonance assignment* and *side-chain resonance assignment*, which refers to resonance assignment for backbone or side-chain atoms. A typical approach for backbone resonance assignment is to exploit the connectivity information in an NMR spectrum that measures the bond interactions between backbone atoms in the main-chain of the primary sequence. For instance, in [1] a globally-consistent Hamiltonian path from an NMR interaction graph is found to align to the primary sequence and obtain backbone resonance assignments. On the other hand, side-chain resonances are normally assigned by exploiting the chemical shift pattern and the through-bond connectivity information in side-chains from an *HCCH total correlation spectroscopy* (HCCH-TOCSY) spectrum, which links up the side-chain resonances with the pre-determined backbone resonances using sequential connectivities. The Biological Magnetic Resonance Bank (BMRB) [59] has collected statistics on observed chemical shifts of all amino acids from a large database of solved protein structures. We call this information the *BMRB statistical information*. This information is often used to assist both backbone and side-chain resonance assignment. Once the correspondence between chemical shifts and atoms in the primary sequence has been identified after resonance assignment, each NOESY cross peak can be assigned to a pair of protons that are potentially correlated via a through-space NOE interaction. This process is called *NOE assignment*. In practice, neither resonance assignment nor NOE assignment is an easy task, since NMR spectra are often complicated by spectral artifacts, missing peaks, experimental noise and peak overlap. The completion of the NOE assignment process immediately provides a set of NOE distance restraints between spatially-neighboring protons, and enables structure calculation software, such as XPLOR-NIH [55] and CYANA [23], to compute the 3D structure of the protein. Besides NOE distance restraints, other NMR geometric constraints can be also used in structure determination. For example, residual dipolar couplings (RDCs) provide global orientational restraints on the internuclear bond vectors [58, 57], and can be also used in structure determination [58, 17, 53, 51, 13, 61, 62, 66].

Although substantial progress has been made in automated backbone resonance assignment [68, 2, 10, 37, 14, 64, 1, 31, 60], general approaches for automated side-chain resonance assignment are still not well developed [43, 48, 5]. Generally the side-chain resonance assignment problem is much more challenging than the backbone resonance assignment problem [48, 5, 47]. Traditional approaches for side-chain resonance assignment [40, 41, 46, 47] usually require a combination of several insensitive side-chain NMR experiments, including HCCH-TOCSY experiments, to obtain enough side-chain resonance assignments. Unfortunately, the performance of HCCH-TOCSY experiments is limited on large proteins due to the fast transverse relaxation of protonated carbons, which causes severe signal loss in NMR spectra. In addition, most large proteins must be deuterated (i.e., most aliphatic protons are replaced with deuterium isotope, and NMR signals from these atoms are muted), to reduce peak overlap and congestion in NMR spectra. The deuteration is also required to increase the efficiency of the *transverse relaxation-optimized spectroscopy* (TROSY) experiments that are generally used to enhance the sensitivity of NMR spectra. The deuteration for large proteins also drastically reduces the number of the NMR-active protons attached to side-chain carbons, which further limits the utility of TOCSY experiments, and thus makes it difficult to attain complete side-chain resonance assignments. On the other hand, it is essential to obtain almost all of the side-chain resonance assignments as a prerequisite for high-resolution structure determination, since they enable the NOE assignment, which constrains side-chain conformations geometrically, thereby enabling high-resolution structure determination. Although new techniques based on high-dimensional NMR experiments have been proposed to overcome the peak overlap issue in side-chain resonance assignment [25, 16], they still incur a penalty in absolute sensitivity. In general, it takes weeks or even months for traditional NMR approaches to collect all these required experimental data, and obtain a nearly complete set of side-chain resonance assignments.

In this paper, we describe a novel algorithm that assigns side-chain resonances from NOESY, backbone chemical shift and RDC data rather than from TOCSY spectra. We cast the side-chain resonance assignment problem into a Markov Random Field (MRF) framework, and apply combinatorial protein design algorithms to compute the optimal solution. Our MRF captures the contact map information in the backbone conformations determined from RDCs using our recently-developed techniques [61, 62, 13, 66], and a set of discretized side-chain conformations (i.e., rotamers) obtained from a high-resolution structure database. A Hausdorff-based computation is incorporated in the scoring function to compute the probability of side-chain resonance assignments to generate the observed NOESY spectra. The optimal side-chain resonance assignments are computed using the following protein design algorithms [12, 45, 19, 18, 9]. First, a *dead-end elimination* (DEE) algorithm [12, 45, 19] is applied to prune side-chain resonance assignments that are *provably* not part of the optimal solution. Second, an A\* search algorithm is employed to find a set of optimal side-chain resonance assignments that best interpret the NMR data. Note that MRF and other graphical models have been used in structural and computational biology. Often they are used with techniques such as belief propagation, which can only be proven to compute a local optimum for a general graph. In contrast, we use DEE and A\* algorithms to provably compute the global optimal solution to the MRF.

In [66], we proposed a high-resolution structure determination approach using an RDC-defined backbone conformation and a pattern-matching technique. Unlike the algorithm in [66] and other previous structure calculation approaches [22, 24, 44, 26, 34], all of which require a nearly complete set of both side-chain and backbone resonance assignments, in this paper the high-resolution structure determination strategy encoded by our algorithm only needs backbone resonance assignments, and does not require any TOCSY-like experiments. Such an advantage can help structural biologists reduce both experimental cost and NMR instrument time, and hence speed up the NMR structure determination process. The following contributions are made in this paper:

1. Introduction of a novel side-chain resonance algorithm that only requires NOESY spectra, backbone chemical shifts, and RDCs, and does not require any TOCSY-like experiments;
2. Development of an MRF framework for side-chain resonance assignment, which captures the contact map information of the protein derived from NOESY spectra, and exploits the structural information inferred from orientational restraints and side-chain rotamers;
3. Introduction of a Hausdorff-based measure to compute a probability distribution of side-chain resonance assignments in the MRF framework;
4. Application of protein design algorithms, including the DEE and A\* search algorithms, to solve the side-chain resonance assignment problem; and
5. Testing and excellent results on real NMR spectra for five proteins recorded at Duke University.

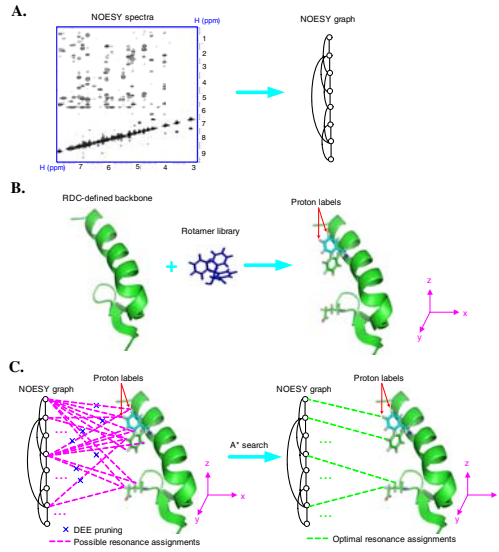
## 2 Methods

### 2.1 Backbone Structure Determination from Residual Dipolar Couplings

We apply our recently-developed algorithms [61, 62, 66, 13] to compute the protein backbone structures using two RDCs per residue (either NH RDCs measured in two media, or NH and CH RDCs measured in a single medium). Details on backbone structure determination from RDCs are available in Supplementary Material [67] **Section 1** and [61, 62, 13]. Alternatively, the global fold (i.e., backbone) could in principle be computed by other approaches, such as protein structure prediction [3], protein threading [65] or homology modeling [35, 36].

### 2.2 Markov Random Field for Side-Chain Resonance Assignment

We introduce notation to describe our side-chain resonance assignment problem. Let  $U = \{r_1, \dots, r_n\}$  be the set of all resonances, including both backbone and side-chain resonances. Here backbone resonances are assigned and taken as input to our algorithm. Side-chain resonances are, of course, unassigned. Let  $t$  be the number of unassigned side-chain resonances, so the number of assigned backbone resonances is  $n - t$ . Without loss of generality, let  $V = \{r_1, \dots, r_t\}$  be the set of unassigned side-chain resonances, and let  $U - V = \{r_{t+1}, \dots, r_n\}$  be the set of assigned backbone resonances.



**Fig. 1.** Schematic illustration of our side-chain resonance assignment algorithm. (A): Construction of the NOESY graph. (B): Construction of proton labels. (C): The side-chain resonance assignment process.

A graph  $G = (U, E)$ , called the *NOESY graph* [2, 1], represents the *contact map* information of resonances from NOESY spectra. In a NOESY graph  $G = (U, E)$ ,  $U$  is the set of proton resonances (including both assigned backbone and unassigned side-chain proton resonances). Two resonances in  $U$  are connected by an edge in  $E$ , when a NOESY cross peak is observed at the coordinates (within a parameterized error window) of these two resonances (Fig. 1A). Nodes in  $U$  are called the *resonance nodes* (or *resonances*). Given a resonance node  $u$  in a NOESY graph  $G = (U, E)$ ,  $N(u) = \{v \mid (u, v) \in E \text{ and } u, v \in U, u \neq v\}$  is called the *neighborhood* of  $u$ . A *proton label* is defined as a 3-tuple that consists of the *proton name* (e.g., Arg16-H $_{\gamma_2}$ ), the *rotamer index* (e.g., the 3rd rotamer in the rotamer library) and the *proton coordinates* in  $\mathbb{R}^3$ . The set of all proton labels is called the *label set*  $L$  of the NOESY graph. We obtain a discrete and finite label set by considering all possible side-chain rotamer conformations on the RDC-defined backbone (Fig. 1B). Since the backbone has been solved and each side-chain rotamer conformation is rigid, each proton label corresponds to a proton on a particular rotamer after being placed on the backbone (with fixed positions in  $\mathbb{R}^3$  with respect to backbone conformation). In our assignment problem, we aim to find a *map*  $\pi : V \rightarrow L$ , such that the contact map information through the mapped resonance nodes in a NOESY graph optimally interprets NOESY spectra. Given a resonance node  $r_i \in V$  and a map  $\pi$ , we call  $\pi(r_i) \in L$  a *proton label assignment* (or *assignment*) of  $r_i$ . Given a sequence of resonances  $W = (r_1, \dots, r_m)$ , we call the sequence  $(\pi(r_1), \dots, \pi(r_m))$  an *assignment* of  $W$ , where  $\pi(r_i)$  is the assignment of resonance node  $r_i$ .

Unlike previous side-chain resonance assignment algorithms [40, 41, 46, 47, 15], which only assign proton names to resonances, our algorithm computes not only the resonance assignments but also the rotamer assignments, since each proton label contains both the proton name and the rotamer index of this proton. The rotamer assignments included in the proton label assignments yield an ensemble of side-chain rotamer conformations for each residue, which are unified by the logical “OR” operation. In our algorithm, proton labels are treated as a cloud of unconnected points in  $\mathbb{R}^3$ . This formulation is similar to [20, 21] which uses a spatial proton distribution to represent a gas of unbound and unassigned hydrogen atoms. Unlike in [20, 21], which depends on molecular dynamics to embed the structure from the unassigned proton density, here we exploit the RDC-defined backbone conformations and apply an MRF to compute the correspondence between side-chain resonances and protons. Although the absence of the covalent structure in proton labels may allow resonances to map to the protons on the same side-chain in different rotameric states, we take into account the distance information of the covalent structure when computing the probability of side-chain resonance assignments (Sec. 2.3). In practice, as we will show in Sec. 3 our MRF can compute a high percentage of correct side-chain resonance assignments for accurate structure determination.

Given a NOESY graph, the assignment of each unassigned resonance  $r_i$  only depends on the resonance assignments of its neighborhood  $N(r_i)$  in  $G$ . We can use a Markov Random Field (MRF) model [33] to encode this assignment problem. The assignment of a resonance node  $r_i$  satisfies the following property:

$$\Pr(\pi(r_i) \mid \pi(r_j), i \neq j) = \Pr(\pi(r_i) \mid \pi(r_j), r_j \in N(r_i)), \quad (1)$$

where  $\Pr(\cdot)$  is the probability of an event, and  $N(r_i)$  is the set of resonance nodes adjacent to  $r_i$  in the graph.

According to the Hammersley-Clifford theorem [6], the distribution of an MRF can be written in a closed form. Let  $C$  be a clique in the underlying graph  $G$ , and let  $T_C(\cdot)$  be a *clique potential* [6] that represents the probability of a particular assignment of all resonance nodes in clique  $C$ . Let  $V' = (r_1, \dots, r_t)$  be an ordered sequence of resonances from set  $V = \{r_1, \dots, r_t\}$ . Let  $F = (\pi(r_1), \dots, \pi(r_t))$  be an assignment for the sequence of resonances  $V'$ . By the Hammersley-Clifford theorem, the probability of an assignment  $F$  is defined by  $\Pr(F) \propto \exp(-\sum_C T_C(F))$ . We consider the potential function  $T_C$  for cliques of size 2, that is, the clique potential involves pairs of neighboring resonance nodes in  $G$ . Note that MRFs with cliques of size of 2 have been widely applied in several areas such as computer vision [8] and computational biology [32, 63]. In our MRF,  $\Pr(F)$  measures the distribution of side-chain resonance assignments by capturing the pairwise resonance interactions in NOESY spectra and exploiting the structural information available from the RDC-defined backbone conformations and the discretized side-chain rotamer conformations.

Given two proton labels with the distance between their coordinates less than 6 Å, we expect to observe an NOE peak in NMR spectra. Such an expected peak is called a *back-computed NOE peak*. In contrast, an NOE peak that has been observed in experimental (NOESY) spectra is called an *experimental NOE peak*. A *back-computed NOE pattern* is defined as a set of back-computed NOE peaks. Since each proton label consists of



the proton name, the rotamer index and the discrete coordinates of the rotamer's side-chain proton, the assignments of a resonance  $r_i$  and its neighborhood  $N(r_i)$  determine a back-computed NOE pattern. A back-computed NOE pattern is constructed as follows. Let  $d(\pi(r_i), \pi(r_j))$  be the Euclidean distance between two proton labels  $\pi(r_i)$  and  $\pi(r_j)$ . Let  $I_{ij} = c \cdot (d(\pi(r_i), \pi(r_j)))^{-6}$  be the back-computed peak intensity using distance  $d(\pi(r_i), \pi(r_j))$ , where  $c$  is the calibration constant that can be computed using the same strategy as in [49,34]. Let  $\lambda(r_i)$  be the resonance of the heavy atom that is covalently bound to the proton corresponding to resonance  $r_i$ . Given a pair of assignments  $\pi(r_i)$  and  $\pi(r_j)$ , we call  $b_{ij}(\pi(r_i), \pi(r_j)) = (r_i, \lambda(r_i), r_j, I_{ij})$  the *back-computed NOE peak* of  $\pi(r_i)$  and  $\pi(r_j)$ . The definitions of back-computed NOE peaks here and experimental NOE peaks in Sec. 2.3 are presented for 3D NOESY spectra. They can be easily extended to other dimensional cases (e.g., 4D). When  $d(\pi(r_i), \pi(r_j))$  is larger than the NOE cutoff 6 Å or two proton labels represent the same proton name, the back-computed NOE peak is a null point. Given a set of resonances  $W \subset U$  and the assignment  $\pi$ , let  $B_\pi(W) = \{b_{ij}(\pi(r_i), \pi(r_j)) | r_i, r_j \in W, r_i \neq r_j\}$  be the *back-computed NOE pattern* of  $W$ .

In our MRF formulation, the clique potential for node  $r_i$  and its neighborhood  $N(r_i)$  can be measured by the matching score of their back-computed NOE pattern. Specifically, let  $V_i = \{r_i\} \cup N(r_i)$ , and let  $B_\pi(V_i)$  be the back-computed NOE pattern of  $V_i$  under the assignment  $\pi$ . Without ambiguity, we will use  $B_i$  to represent  $B_\pi(V_i)$ . Let  $s(B_i)$  be the matching score of the back-computed NOE pattern  $B_i$ , where the function  $s(\cdot)$  will be defined in Sec. 2.3. We use  $T_\pi(r_i, N(r_i)) = -s(B_i)$  to represent the clique potential of the pairwise interactions between  $r_i$  and its neighborhood  $N(r_i)$ . Thus, we have the following function for the probability of an MRF  $F = (\pi(r_1), \dots, \pi(r_t))$ :

$$\Pr(F) \propto \exp\left(-\sum_{r_i \in V} T_\pi(r_i, N(r_i))\right) = \exp\left(\sum_{r_i \in V} s(B_i)\right). \quad (2)$$

We use  $Q$  to represent the BMRB statistical information (see Sec. 1). To estimate the probability of an MRF  $F$  based on the BMRB statistical information  $Q$ , we first relate them using the probability function  $\Pr(Q|F)$ . Recall that  $\lambda(r_i)$  represents the frequency of the heavy atom covalently bound to the proton corresponding to  $r_i$ . The probability function  $\Pr(Q|F)$  is defined by

$$\Pr(Q|F) = \prod_{r_i \in V} P(|r_i - \mu_i|, \sigma_i) \cdot P(|\lambda(r_i) - \mu'_i|, \sigma'_i), \quad (3)$$

where  $P(|x - \mu|, \sigma)$  is the probability of observing the difference  $|x - \mu|$  in a normal distribution with mean  $\mu$  and standard deviation  $\sigma$ , and  $\mu_i, \sigma_i, \mu'_i, \sigma'_i$  are average values and standard deviations of chemical shifts derived from BMRB given the assignment  $\pi(r_i)$ . We note that the normal distribution and other similar distribution families have been widely used to model the noise in the NMR data, e.g., see [52] and [38].

By Bayes' Rule,  $\Pr(F|Q)$ , the probability of the assignment  $F$  conditioned on the BMRB statistical information  $Q$  (namely the *posterior probability*), can be computed as follows:

$$\Pr(F|Q) \propto \Pr(F) \cdot \Pr(Q|F) \quad (4)$$

$$\propto \exp\left(-\sum_{r_i \in V} T(\pi(r_i), \pi(N(r_i)))\right) \cdot \prod_{r_i \in V} P(|r_i - \mu_i|, \sigma_i) \cdot P(|\lambda(r_i) - \mu'_i|, \sigma'_i) \quad (5)$$

$$= \exp\left(\sum_{r_i \in V} s(B_i)\right) \cdot \prod_{r_i \in V} P(|r_i - \mu_i|, \sigma_i) \cdot P(|\lambda(r_i) - \mu'_i|, \sigma'_i). \quad (6)$$

Our goal is to compute an assignment  $F^* = (\pi^*(r_1), \dots, \pi^*(r_t))$  that maximizes the posterior probability  $\Pr(F|Q)$ . Taking the negative logarithm on both sides of Eq. (6), we have the following *pseudo-energy* function for an assignment  $F = (\pi(r_1), \dots, \pi(r_t))$ :

$$E_F = -\sum_{r_i \in V} \ln P(|r_i - \mu_i|, \sigma_i) \cdot P(|\lambda(r_i) - \mu'_i|, \sigma'_i) - \sum_{r_i \in V} s(B_i). \quad (7)$$

The pseudo-energy function in Eq. (7) measures how well an assignment  $F = (\pi(r_1), \dots, \pi(r_t))$  satisfies both the BMRB statistical information and the experimental NMR data. Maximizing the posterior probability  $\Pr(F|Q)$  in Eq. (6) is equivalent to minimizing the pseudo-energy function in Eq. (7). We call the assignment  $F^* = (\pi^*(r_1), \dots, \pi^*(r_t))$ , that minimizes the scoring function  $E_F$  and thus best interprets the NMR data restraints, the *optimal assignment* or *optimal solution* to our MRF. Since our proton label assignments contain both resonance assignments and molecular side-chain coordinates, the optimal assignment is analogous to the *global minimum energy conformation* (GMEC) in the protein design literature.

### 2.3 The Matching Score of a Back-Computed NOE Pattern

The *matching score* of a back-computed NOE pattern can be measured by comparing the back-computed peaks with NOESY spectra. Given a set of resonance nodes  $W \subset U$  and an assignment  $\pi$ , let  $B_\pi(W)$  denote their back-computed NOE pattern. Without ambiguity, we will use  $B$  to stand for  $B_\pi(W)$ . Let  $Y$  be the set of experimental peaks. The matching score between the back-computed NOE pattern  $B$  and experimental spectrum  $Y$  can be measured by the conventional Hausdorff distance  $H(B, Y) = \max(h(B, Y), h(Y, B))$ , where  $h(B, Y) = \max_{b \in B} \min_{y \in Y} \|b - y\|$  and  $\|\cdot\|$  is the normed distance. This conventional Hausdorff distance is sensitive to a single outlying point of  $B$  or  $Y$  [28, 29]. For example, suppose that an NOE peak is missing in  $Y$  (which is quite common in NMR data), and its corresponding back-computed peak in  $B$  has a large distance from any peak in  $Y$ . In such a case, the Hausdorff distance between  $B$  or  $Y$  is dominated by this missing NOE peak. To take into account the missing NOE peaks, we employ a generalized Hausdorff distance measure, called the *Hausdorff fraction* (*fractional Hausdorff distance*), which is derived from the  $k^{\text{th}}$  Hausdorff distance  $h_k$  from  $B$  to  $Y$  [29, 27]:

$$h_k(B, Y) = k^{\text{th}} \min_{b \in B} \min_{y \in Y} \|b - y\|,$$



where  $k^{th}$  is the  $k^{th}$  largest value. Now, let  $\delta$  be the error window in chemical shift. Then the probability of the back-computed NOE pattern  $B$  under  $h_k(B, Y) \leq \delta$ , is computed by the following Hausdorff fraction equation [27]:

$$s(B) = \frac{\tau(B \cap Y_\delta)}{\tau(B)}, \quad (8)$$

where  $Y_\delta$  denotes the union of all balls obtained by replacing each point in  $Y$  with a ball of radius  $\delta$ , and  $\tau(\cdot)$  denotes the size of a set.

Next, we will show how to compute the matching score of a back-computed NOE pattern in Eq. (8). Let  $b_{ij}(\pi(r_i), \pi(r_j)) = (r_i, \lambda(r_i), r_j, I_{ij})$  be a back-computed NOE peak in  $B$  based on assignments  $\pi(r_i)$  and  $\pi(r_j)$ , where  $\lambda(r_i)$  is the frequency of the heavy atom covalently bound to the proton corresponding to  $r_i$ , and  $I_{ij}$  is the back-computed peak intensity. Without ambiguity, we will use  $b_{ij}$  to represent  $b_{ij}(\pi(r_i), \pi(r_j))$ . Note that the distance information of the covalent structure is also included when computing a back-computed NOE pattern, since the distances between protons within a residue or in consecutive residues are generally  $< 6 \text{ \AA}$ . Let  $(x, y, z, I')$  be the experimental NOESY cross peak that is closest to the back-computed NOE peak  $b_{ij}$  under the Euclidean distance measure, where  $x$  and  $z$  are frequencies of NOE interacting protons,  $y$  is the frequency of the heavy atom covalently bound to the first proton, and  $I'$  is the peak intensity. When computing the geometric count  $\tau(B \cap Y_\delta)$ , we must take into account the uncertainty in chemical shift. For example, suppose that the back-computed NOE peak  $b_{ij}$  is within the Euclidean distance  $\delta$  from an experimental NOESY cross peak. When  $b_{ij}$  is closer to this experimental peak, it should contribute more to counting  $\tau(B \cap Y_\delta)$ . To measure the probability of a back-computed NOE peak to intersect with  $Y_\delta$ , we model the uncertainty of chemical shifts in individual dimensions as independent normal distributions. Formally, the following equation is employed to compute  $\tau(B \cap Y_\delta)$ :

$$\tau(B \cap Y_\delta) = \sum_{b_{ij} \in B} P(|I' - I_{ij}|, \sigma_{I\delta}) \cdot P(|x - r_i|, \sigma_{x\delta}) \cdot P(|y - \lambda(r_i)|, \sigma_{y\delta}) \cdot P(|z - r_j|, \sigma_{z\delta}), \quad (9)$$

where  $P(|x - \mu|, \sigma)$  is the probability of observing the difference  $|x - \mu|$  in a normal distribution with mean  $\mu$  and standard deviation  $\sigma$ . We define the standard deviations in Eq. (9) as a function of the error window  $\delta$ . We choose  $\sigma = \delta/3$  for each dimension such that the probability for a back-computed NOE peak outside  $Y_\delta$  to contribute to  $\tau(B \cap Y_\delta)$  is almost 0.

## 2.4 A DEE Pruning Algorithm

The chemical shift of each proton in a particular residue usually lies within an interval derived from the BMRB statistical information [59]. Therefore, each resonance node  $r_i$  in the NOESY graph is only allowed to map to a subset of proton labels, in which the BMRB-derived chemical shift intervals contain the frequency of  $r_i$ . Given a resonance  $r_i$ , we call the subset of proton labels in  $L$ , that  $r_i$  is allowed to map to, the *candidate*

mapping set of  $r_i$ , denoted by  $A(r_i)$ . When we know the backbone resonance assignments, we have  $|A(r_i)| = 1$  for all backbone resonance nodes  $r_i$ . Given a sequence of resonances  $W = (r_1, \dots, r_m)$ , we call  $A(W) = (A(r_1), \dots, A(r_m))$  the *candidate mapping set* of  $W$ . Let  $D = (\pi(r_1), \dots, \pi(r_m))$ , where  $\pi(r_i) \in A(r_i)$  is the assignment of  $r_i$ . We write  $D \in A(W)$  when  $\pi(r_i) \in A(r_i)$  for every  $i = 1, \dots, m$ , i.e., the assignment of  $r_i$  lies in the candidate mapping sets for all resonances.

We use  $\gamma(r_i, u)$  to mean that proton label  $u \in L$  is assigned to resonance node  $r_i$ , where  $u \in A(r_i)$ . Initially, we prune any proton label assignment  $\gamma(r_i, u)$  in which the frequency of  $r_i$  falls outside the BMRB-derived chemical shift interval. Let  $N(r_i) = \{r'_{i1}, \dots, r'_{im}\}$  be the set of resonance nodes in the neighborhood of  $r_i$ , and let  $N'(r_i) = (r'_{i1}, \dots, r'_{im})$  be a sequence of resonance nodes in  $N(r_i)$ , where  $m$  is total number of resonance nodes in the neighborhood. Then the candidate mapping set of  $N'(r_i) = (r'_{i1}, \dots, r'_{im})$  is  $A(N'(r_i)) = (A(r'_{i1}), \dots, A(r'_{im}))$ . Let  $D_i = (\pi(r'_{i1}), \dots, \pi(r'_{im})) \in A(N'(r_i))$  be an assignment of  $N'(r_i)$ , where  $\pi(r'_{ij}) \in A(r'_{ij})$ , and we use  $\gamma(N'(r_i), D_i)$  to mean that  $D_i$  is assigned to  $N'(r_i)$ .

Given an assignment  $F = (\pi(r_1), \dots, \pi(r_t))$  for the sequence of resonances  $V' = (r_1, \dots, r_t)$ , we use  $E(\gamma(r_i, \pi(r_i)))$  to represent the first energy term in Eq. (7) under the assignment  $\pi$ . We use  $E(\gamma(r_i, \pi(r_i)), \gamma(N'(r_i), D_i))$  to represent the second energy term in Eq. (7) when assigning  $\pi(r_i)$  to resonance node  $r_i$  and  $D_i$  to  $N'(r_i)$ , where  $\pi(r_i) \in A(r_i)$  and  $D_i \in A(N'(r_i))$ . Then the pseudo-energy scoring function in Eq. (7) for an assignment  $F = (\pi(r_1), \dots, \pi(r_t))$  can be rewritten as

$$E_F = \sum_{r_i \in V} E(\gamma(r_i, \pi(r_i))) + \sum_{r_i \in V} E(\gamma(r_i, \pi(r_i)), \gamma(N'(r_i), D_i)), \quad (10)$$

where  $\pi(r_i) \in A(r_i)$  and  $D_i \in A(N'(r_i))$ .

An algorithm that is similar to the GMCC calculation method in protein design [12, 45, 19, 18, 9] can be applied here to compute the optimal proton label assignments. The *dead-end elimination* (DEE) algorithm has been effectively applied to prune rotamers when their contribution to the total energy is always less than another (competing) rotamer [12, 45, 19, 18, 9]. We use a similar idea here to prune proton label assignments that are *provably* not part of the optimal solution. Given an unassigned side-chain resonance node  $r_i \in V$ , a proton label assignment  $v \in A(r_i)$  is eliminated if an alternative proton label assignment  $u \in A(r_i)$  satisfies the following Goldstein criterion [19]:

$$E(\gamma(r_i, v)) - E(\gamma(r_i, u)) + \min_{D_i \in A(N'(r_i))} \left( E(\gamma(r_i, v), \gamma(N'(r_i), D_i)) - E(\gamma(r_i, u), \gamma(N'(r_i), D_i)) \right) > 0. \quad (11)$$

Any assignment  $\gamma(r_i, v)$  satisfying Eq. (11) is *provably* not part of the optimal solution, and thus can be safely pruned. The complexity of computing the Goldstein criterion in Eq. (11) is  $O(na^2w)$ , where  $n$  is the total number of resonances,  $a$  is the maximum number of proton labels in the candidate mapping set of a resonance, and  $w$  is the maximum number of proton labels that can be assigned to a resonance node's neighborhood. DEE reduces the conformation search space by pruning proton label assignments that can not be in the optimal solution, and provides a combinatorial factor reduction in computational complexity.

## 2.5 Computing Optimal Side-Chain Resonance Assignments

To compute the optimal solution to our MRF, we apply an A\* algorithm [39, 54, 56] to search over all possible combinations of the remaining proton label assignments surviving from DEE. An A\* algorithm provably finds the optimal (i.e., least-cost) path from a given starting node to the goal node in a search tree or graph. It uses a heuristic cost function to determine the order of visiting nodes during the search. The heuristic cost function consists of two parts: the *actual* cost from the starting node to the current node, and the *estimated* cost from the current node to the goal node. Next, we will define both actual and estimated cost functions that are used to determine the order of searching nodes in our A\* algorithm.

Recall that  $V' = (r_1, \dots, r_t)$  denotes the sequence of unassigned side-chain resonances, and  $(r_{t+1}, \dots, r_n)$  denotes the sequence of assigned backbone resonances. Let  $X_i$  be the variable representing the assignment of resonance node  $r_i$ . Similar to the protein design problem [39, 18], our search configuration space can be also formulated as a tree, in which the root represents an empty assignment, a leaf node represents a full assignment of  $V'$ , and an internal node represents a partial assignment of  $V'$  (i.e., only a subsequence of resonances in  $V'$  are assigned). Let  $H = (X_{t+1}, \dots, X_n)$  be the sequence of known assignments for backbone resonances  $(r_{t+1}, \dots, r_n)$ . Let  $S = (X_1, \dots, X_t)$  be a sequence of assignments for side-chain resonances in  $V'$ . Given the BMRB statistical information  $Q$  and the known backbone chemical shifts  $H$ , the probability for a sequence of side-chain resonance assignments  $S$  is

$$\Pr(S|H, Q) = \Pr(X_t, X_{t-1}, \dots, X_1|H, Q) = \Pr(X_t|X_{t-1}, \dots, X_1, H, Q) \cdots \Pr(X_2|X_1, H, Q) \cdot \Pr(X_1|H, Q). \quad (12)$$

Suppose that the A\* algorithm has assigned resonances  $r_1, \dots, r_{i-1}$ . We rewrite Eq. (12) as

$$\Pr(S|H, Q) = \Pr(X_t|X_{t-1}, \dots, X_1, H, Q) \cdots \Pr(X_{i+1}|X_i, \dots, X_1, H, Q) \cdot \Pr(X_i|X_{i-1}, \dots, X_1, H, Q) \cdots \Pr(X_1|H, Q). \quad (13)$$

Taking the negative logarithm on both sides of Eq. (13), we have

$$\begin{aligned} -\ln \Pr(S|H, Q) &= -\ln(\Pr(X_t|X_{t-1}, \dots, X_1, H, Q) \cdots \Pr(X_{i+1}|X_i, \dots, X_1, H, Q)) \\ &\quad -\ln(\Pr(X_i|X_{i-1}, \dots, X_1, H, Q) \cdots \Pr(X_1|H, Q)). \end{aligned} \quad (14)$$

Eq. (14) measures the *cost* of a path from the root (i.e., empty assignment) to one of leaf nodes (i.e., full assignments) in our A\* search tree.

Let

$$g = -\ln(\Pr(X_i|X_{i-1}, \dots, X_1, H, Q) \cdots \Pr(X_1|H, Q)), \quad (15)$$

which measures the probability of the set of the first  $i$  assignments  $X_1, \dots, X_i$ , and leads to the actual cost of the path from the root to the current node in the A\* search tree.

Let

$$h = -\ln(\Pr(X_t|X_{t-1}, \dots, X_1, H, Q) \cdots \Pr(X_{i+1}|X_i, \dots, X_1, H, Q)), \quad (16)$$

which estimates the cost of assigning the remaining resonance nodes (i.e., the cost of the path from current node to the leaf nodes in the A\* search tree).

Then the cost function in our A\* search is defined by

$$f = g + h, \quad (17)$$

where  $g$  is the *actual cost* from the root to the current node in the search tree, and  $h$  is the *estimated cost* from the current node to one of leaf nodes, in which all side-chain resonances are assigned.

In Eq. (16),  $\Pr(X_j|X_{j-1}, \dots, X_i, \dots, X_1, H, Q)$ ,  $j > i$ , is estimated as follows:

$$\begin{aligned} & \Pr(X_j|X_{j-1}, \dots, X_i, \dots, X_1, H, Q) \\ = & \max_{\substack{u_j \in A(r_j) \\ \dots \\ u_{i+1} \in A(r_{i+1})}} \Pr(\gamma(r_j, u_j)|\gamma(r_{j-1}, u_{j-1}), \dots, \gamma(r_{i+1}, u_{i+1}), X_i, \dots, X_1, H, Q), \end{aligned} \quad (18)$$

where  $\gamma(r_j, u_j)$  denotes the assignment of  $u_j$  to resonance node  $r_j$ .

The A\* algorithm maintains a list of search nodes, which are ranked according to the cost function (Eq. (17)). Similar to the protein design work in [18], here the A\* search algorithm expands the nodes in order of the cost function  $f$ . In each iteration, the node with the smallest  $f$  value is visited, and the values of  $f$  in the remaining nodes are updated. All remaining nodes in the list are re-ordered according to the new  $f$  values, and form the children of current visited node. Such a process is repeated until all side-chain resonances are assigned (i.e., when a leaf node in the search tree is reached).

An estimated cost function is *admissible*, if it does not overestimate the cost from any node to the goal node. The admissibility of the estimated cost function ensures that an A\* search algorithm will find the optimal solution. The following claim provides the soundness of our A\* algorithm in computing the optimal assignment. The proof of this claim is provided in Supplementary Material **Section 2** available online in Ref. [67].

**Claim 1.** *The estimated cost function defined in Eq. (18) is admissible, which guarantees that our A\* search algorithm will find the optimal solution.*

The A\* algorithm is proven to be complete and optimal in searching for the least-cost path [39,54,56]. Although the time complexity of the A\* algorithm is exponential in the number of side-chain resonances in the worst case, in practice, our algorithm, including both DEE and A\* modules, runs only in hours for a medium-size protein. For instance, it takes about 7 hours to compute the set of side-chain resonance assignments on a single-processor machine for the human ubiquitin protein without human intervention.

### 3 Results

We have tested our algorithm on NMR data of five proteins: the FF Domain 2 of human transcription elongation factor CA150 (FF2), the B1 domain of Protein G (GB1),

human ubiquitin, the ubiquitin-binding zinc finger domain of the human Y-family DNA polymerase Eta (pol  $\eta$  UBZ), and the human Set2-Rpb1 interacting domain (hSRI). The numbers of amino acid residues in these proteins are 62 for FF2, 39 for pol  $\eta$  UBZ, 56 for GB1, 76 for ubiquitin, and 112 for hSRI. Note that by the standards of the NMR community [22, 5, 24, 55, 23, 64, 52, 60], tests on real experimental data of five proteins are sufficient to demonstrate the feasibility of an algorithm in NMR data analysis and structure determination. All NMR data except RDCs of ubiquitin and GB1 were recorded and collected using Varian 600 and 800 MHz spectrometers at Duke University. The NOE cross peaks were picked from 3D  $^{15}\text{N}$ - and  $^{13}\text{C}$ -edited NOESY-HSQC spectra. The NH and CH RDC data of FF2, pol  $\eta$  UBZ and hSRI were measured from a 2D  $^1\text{H}$ - $^{15}\text{N}$  IPAP experiment [50] and a modified (HACACO)NH experimental [4] respectively. Details on the NMR experimental procedures and results on the backbone structure calculation from RDCs are provided in Supplementary Material **Section 3** available online in Ref. [67].

### 3.1 Accuracy of Side-Chain Resonance Assignments

We evaluated the accuracy of the side-chain resonances assigned by our algorithm by comparing them with the chemical shifts of the proteins that were assigned manually using other additional side-chain NMR experiments. Our algorithm achieved the completeness of over 90% for resonance assignments, that is, it assigned the resonances of over 90% of protons (Table I). Note that the manual assignments are usually obtained from TOCSY experiments, while frequencies in our resonance list are extracted from NOESY spectra. Due to the experimental uncertainty, frequencies of our assigned resonances are not exactly equal to the manually-assigned chemical shifts. We used an error window 0.04 ppm for  $^1\text{H}$ , and 0.4 ppm for heavy atoms (i.e.,  $^{13}\text{C}$  and  $^{15}\text{N}$ ) to check whether two resonance assignments agree with each other. We say a resonance assignment is *correct* if its frequency is within the error window from the reference assignment, which was assigned manually using other additional experiments. Our tests show that our algorithm computes about 80% of the correct resonance assignments (Table I).

**Table 1.** Summary of side-chain resonance assignment results

Proteins	GB1	ubiquitin	hSRI	pol $\eta$ UBZ	FF2
Completeness (%)	97.7	94.9	90.2	97.6	92.7
Correctness (%)	81.9	81.8	83.6	92.2	78.0

In a hypothetical ideal case without any experimental error and noise, the goal of an NMR assignment problem is to find a one-to-one correspondence (i.e., bijection) between resonances and proton names in the protein sequence. In practice, a proton can be mapped to 2-3 different resonances due to the ambiguity arising from chemical shift degeneracy, that is, chemical shifts of two different protons may be so close that the probabilities measuring their assignments are not sufficient to distinguish them. In practice, the optimal solution to our MRF finds the one-to-one mapping for most resonance assignments (Table I), because the local neighborhood structure of our MRF has

enforced these correct assignments. Most of the *inconsistent* assignments (i.e., two resonances are assigned to the same proton label) occur in the methylene protons bound to the same carbon, or neighboring ring protons in aromatic residues. These protons often have both similar chemical shifts and close coordinates in  $\mathbb{R}^3$ , which makes it difficult to distinguish them using the probability functions derived from our MRF framework. We use the Boolean operation “XOR” to unify these inconsistent assignments. As we will show in Sec. 3.2, the NOE assignment ambiguity arising from these inconsistent resonance assignments does not degrade high-resolution structure determination, probably because these protons are adjacent in  $\mathbb{R}^3$  (with distance  $< 1.8\text{--}2.5$  Å).

### 3.2 Effectiveness for High-Resolution Structure Determination

To investigate the effect of assigned side-chain resonances on high-resolution structure determination, we first computed a set of NOE assignments using the side-chain resonance assignments computed by our algorithm. We then examined the quality of the structures calculated using these NOE distance restraints. Details on computing NOE distance restraints using assigned side-chain resonances are provided in Supplementary Material **Section 4** available online in Ref. [67].

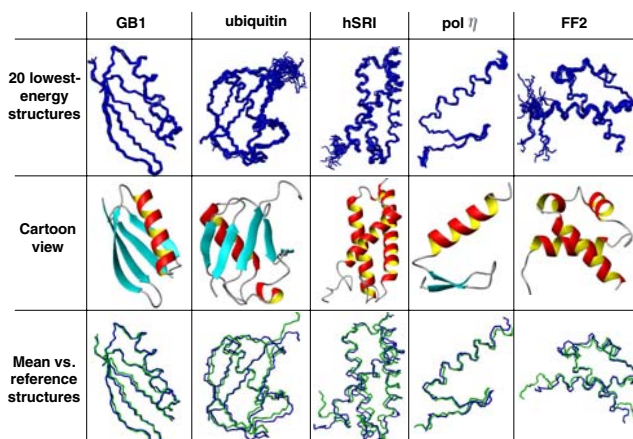
To examine the accuracy of the NOE assignments computed by our algorithm, we compared them with the reference structures. We say an NOE assignment is *correct* if it agrees with the reference structure, that is, the distance between the assigned pair of NOE protons in the reference structure satisfies the NOE restraint whose distance is calibrated from the experimental peak intensity. As shown in Table 2, our algorithm computes over 80% correct NOE restraints. To further investigate these NOE distance restraints, we fed them into XPLOR-NIH [55] for the structure calculation. To fairly compare the accuracy of our NOE restraints, we fed the same hydrogen bond and dihedral angle constraints into XPLOR-NIH, as in computing the NMR reference structures. In addition, the structures were refined with RDC data using XPLOR-NIH with a water-refinement protocol [55]. We chose the ensemble of top 20 structures with the lowest

**Table 2.** Summary of NOE assignment results

Proteins	GB1	ubiquitin	hSRI	pol $\eta$ UBZ	FF2
<b>Total # of assigned NOEs</b>	1421	1531	3540	960	1354
Intraresidue	597	648	1326	419	619
Sequential ( $ i - j  = 1$ )	295	321	777	254	282
Medium-range ( $ i - j  \leq 4$ )	185	202	984	177	281
Long-range ( $ i - j  \geq 5$ )	344	360	453	110	172
<b>Percentage of correct NOE assignments (%)</b>	87.0	81.7	83.3	89.4	85.5

**Table 3.** Summary of final calculated structures

Proteins	GB1	ubiquitin	hSRI	pol $\eta$ UBZ	FF2
<b>Average RMSD to mean coordinates</b>					
SSE region (backbone, heavy) (Å)	0.18, 0.38	0.36, 0.71	0.29, 0.75	0.12, 0.43	0.25, 0.67
Ordered region (backbone, heavy) (Å)	0.20, 0.41	0.58, 0.95	0.35, 0.81	0.15, 0.67	0.34, 0.89
<b>RMSD to reference structure</b>					
SSE region (backbone, heavy) (Å)	0.56, 1.14	0.63, 1.40	1.25, 1.93	0.62, 1.39	0.58, 1.53
Ordered region (backbone, heavy) (Å)	0.54, 1.08	0.93, 1.51	1.37, 2.09	0.97, 1.73	1.06, 2.17



**Fig. 2.** Final NMR structures computed using our automatically-assigned NOEs. Row 1: the ensemble of 20 lowest-energy NMR structures. Row 2: ribbon view of one structure in the ensemble. Row 3: backbone overlay of the mean structures (blue) vs. corresponding NMR reference structures (green) (PDB ID of GB1 [30]: 3GB1; PDB ID of ubiquitin [11]: 1D3Z; PDB ID of FF2: 2E71; PDB ID of hSRI [42]: 2A7O; PDB ID of pol  $\eta$  UBZ [7]: 2I5O).

energies out of 50 structures computed by XPLOR-NIH as the ensemble of final structures. For all five proteins, the ensemble of top 20 structures with the lowest energies converge into a compact cluster (Table 3 and Fig. 2). The average RMSD to the mean coordinates is  $\leq 0.6$  Å for backbone atoms and  $\leq 1.0$  Å for all-heavy atoms. We superimposed the mean structure of the ensemble with the reference structure for each protein. The RMSD between the mean structure and the reference structure (ordered region) is 0.5–1.4 Å for backbone atoms and 1.0–2.2 Å for all-heavy atoms (Table 3 and Fig. 2). These results indicate that the NOE assignments computed by our algorithm are sufficient for high-resolution structure determination.

## 4 Conclusions

Side-chain resonance assignments are essential for high-resolution structure determination and side-chain dynamics studies. In this paper we proposed an MRF with protein design algorithms to compute the set of optimal side-chain resonance assignments that best interpret the NMR data. Tests on real NMR data demonstrated that our algorithm computes a high percentage of accurate side-chain resonance assignments for high-resolution structure determination. Since our algorithm does not require any TOCSY-like experiments, it can advance NMR structure determination by saving a significant amount of both experimental cost and NMR instrument time.

In [15], the authors proposed an algorithm that uses the knowledge of local covalent polypeptide structures to iteratively assign side-chain resonances from previously-assigned resonances (initially backbone resonances were assigned) using NOESY or



TOCSY spectra. Compared to [15], in which only the conformation-independent bounds on intra-residue and sequential inter-proton distances are used to iteratively assign side-chain resonances, our algorithm applies an MRF that effectively exploits the RDC-defined backbone conformations to derive side-chain resonance assignments.

Although our algorithm is only implemented for 3D NOESY spectra, it is general and can be easily extended to higher-dimensional NOESY spectra. In addition, it would be interesting to extend our algorithm to perform side-chain resonance assignment without requiring backbone resonance assignments. Because RDCs are mapped to backbone resonances, in this case, we might have to resort to other approaches such as protein structure prediction, protein threading or homology modeling to obtain the initial global fold.

## Availability

The source code of our algorithm is available by contacting the authors, and is distributed open-source under the GNU Lesser General Public License (Gnu, 2002).

## Acknowledgements

We thank Dr. C. Bailey-Kellogg, Dr. M.S. Apaydin and Mr. J. Martin for reading our draft and providing us with valuable comments. We thank all members of the Donald and Zhou Labs for helpful discussions and comments. We are grateful to Ms. M. Bomar for helping us with pol  $\eta$  UBZ NMR data.

## References

1. Bailey-Kellogg, C., Chainraj, S., Pandurangan, G.: A Random Graph Approach to NMR Sequential Assignment. *Journal of Computational Biology* 12(6), 569–583 (2005)
2. Bailey-Kellogg, C., Widge, A., Kelley, J.J., Berardi, M.J., Bushweller, J.H., Donald, B.R.: The NOESY jigsaw: automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. *Journal of Computational Biology* 7(3-4), 537–558 (2000)
3. Baker, D., Sali, A.: Protein structure prediction and structural genomics. *Science* 294, 93–96 (2001)
4. Ball, G., Meenan, N., Bromek, K., Smith, B.O., Bella, J., Uhrin, D.: Measurement of one-bond  $^{13}\text{C}^\alpha\text{-}^1\text{H}^\alpha$  residual dipolar coupling constants in proteins by selective manipulation of  $\text{C}^\alpha\text{H}^\alpha$  spins. *Journal of Magnetic Resonance* 180, 127–136 (2006)
5. Baran, M.C., Huang, Y.J., Moseley, H.N., Montelione, G.T.: Automated analysis of protein NMR assignments and structures. *Chem. Rev.* 104, 3456–3541 (2004)
6. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc. B* 36 (1974)
7. Bomar, M.G., Pai, M., Tzeng, S., Li, S., Zhou, P.: Structure of the ubiquitin-binding zinc finger domain of human DNA Y-polymerase  $\eta$ . *EMBO reports* 8, 247–251 (2007)
8. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 648 (1998)



9. Chen, C.Y., Georgiev, I., Anderson, A.C., Donald, B.R.: Computational structure-based re-design of enzyme activity. *Proc. Natl. Acad. Sci. USA* 106, 3764–3769 (2009)
10. Coggins, B.E., Zhou, P.: PACES: Protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR* 26, 93–111 (2003)
11. Cornilescu, G., Marquardt, J.L., Ottiger, M., Bax, A.: Validation of Protein Structure from Anisotropic Carbonyl Chemical Shifts in a Dilute Liquid Crystalline Phase. *Journal of the American Chemical Society* 120, 6836–6837 (1998)
12. Desmet, J., Maeyer, M.D., Hazes, B., Lasters, I.: The dead-end elimination theorem and its use in protein side-chain positioning. *Nature* 356, 539–542 (1992)
13. Donald, B.R., Martin, J.: Automated NMR assignment and protein structure determination using sparse dipolar coupling constraints. *Progress in NMR Spectroscopy* 55, 101–127 (2009)
14. Eghbalian, H.R., Bahrami, A., Wang, L.Y., Assadi, A., Markley, J.L.: Probabilistic identification of spin systems and their assignments including coil-helix inference as output (PIS-TACHIO). *J. Biomol. NMR* 32, 219–233 (2005)
15. Fiorito, F., Herrmann, T., Damberger, F.F., Wüthrich, K.: Automated amino acid side-chain NMR assignment of proteins using (13)C- and (15)N-resolved 3D [(1)H, (1)H]-NOESY. *J. Biomol. NMR* 42, 23–33 (2008)
16. Fiorito, F., Hiller, S., Wider, G., Wüthrich, K.: Automated resonance assignment of proteins: 6D APSY-NMR. *J. Biomol. NMR* 35, 27–37 (2006)
17. Fowler, C.A., Tian, F., Al-Hashimi, H.M., Prestegard, J.H.: Rapid determination of protein folds using residual dipolar couplings. *Journal of Molecular Biology* 304, 447–460 (2000)
18. Georgiev, I., Lilien, R.H., Donald, B.R.: The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. *Journal of Computational Chemistry* 29, 1527–1542 (2008)
19. Goldstein, R.F.: Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal* 66, 1335–1340 (1994)
20. Grishaev, A., Llinás, M.: CLOUDS, a protocol for deriving a molecular proton density via NMR. *Proc. Natl. Acad. Sci. USA* 99, 6707–6712 (2002)
21. Grishaev, A., Llinás, M.: Protein structure elucidation from NMR proton densities. *Proc. Natl. Acad. Sci. USA* 99, 6713–6718 (2002)
22. Güntert, P.: Automated NMR Protein Structure Determination. *Progress in Nuclear Magnetic Resonance Spectroscopy* 43, 105–125 (2003)
23. Güntert, P.: Automated NMR protein structure calculation with CYANA. *Meth. Mol. Biol.* 278, 353–378 (2004)
24. Herrmann, T., Güntert, P., Wüthrich, K.: Protein NMR Structure Determination with Automated NOE Assignment Using the New Software CANDID and the Torsion Angle Dynamics Algorithm DYANA. *Journal of Molecular Biology* 319(1), 209–227 (2002)
25. Hiller, S., Joss, R., Wider, G.: Automated NMR assignment of protein side chain resonances using automated projection spectroscopy (APSY). *J. Am. Chem. Soc.* 130(36), 12073–12079 (2008)
26. Huang, Y.J., Tejero, R., Powers, R., Montelione, G.T.: A topology-constrained distance network algorithm for protein structure determination from NOESY data. *Proteins: Structure Function and Bioinformatics* 62(3), 587–603 (2006)
27. Huttenlocher, D.P., Jaquith, E.W.: Computing visual correspondence: Incorporating the probability of a false match. In: *Proceedings of the Fifth International Conference on Computer Vision (ICCV 1995)*, pp. 515–522 (1995)

28. Huttenlocher, D.P., Kedem, K.: Distance Metrics for Comparing Shapes in the Plane. In: Donald, B.R., Kapur, D., Mundy, J. (eds.) *Symbolic and Numerical Computation for Artificial Intelligence*, pp. 201–219. Academic Press, London (1992)
29. Huttenlocher, D.P., Klanderma, G.A., Rucklidge, W.: Comparing Images Using the Hausdorff Distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 15(9), 850–863 (1993)
30. Juszewski, K., Gronenborn, A.M., Clore, G.M.: Improving the Packing and Accuracy of NMR Structures with a Pseudopotential for the Radius of Gyration. *Journal of the American Chemical Society* 121, 2337–2338 (1999)
31. Kamisetty, H., Bailey-Kellogg, C., Pandurangan, G.: An efficient randomized algorithm for contact-based NMR backbone resonance assignment. *Bioinformatics* 22(2), 172–180 (2006)
32. Kamisetty, H., Xing, E.P., Langmead, C.J.: Free Energy Estimates of All-atom Protein Structures Using Generalized Belief Propagation. *Journal of Computational Biology* 15, 755–766 (2008)
33. Kindermann, R., Snell, J.L.: *Markov Random Fields and Their Applications*. American Mathematical Society, Providence (1980)
34. Kuszewski, J., Schwieters, C.D., Garrett, D.S., Byrd, R.A., Tjandra, N., Clore, G.M.: Completely automated, highly error-tolerant macromolecular structure determination from multidimensional nuclear overhauser enhancement spectra and chemical shift assignments. *J. Am. Chem. Soc.* 126(20), 6258–6273 (2004)
35. Langmead, C.J., Donald, B.R.: 3D structural homology detection via unassigned residual dipolar couplings. In: *Proceedings of 2003 IEEE Comput. Syst. Bioinform. Conf.*, pp. 209–217 (2003)
36. Langmead, C.J., Donald, B.R.: High-throughput 3D structural homology detection via NMR resonance assignment. In: *Proceedings of 2004 IEEE Comput. Syst. Bioinform. Conf.*, pp. 278–289 (2004)
37. Langmead, C.J., Yan, A.K., Lilien, R.H., Wang, L., Donald, B.R.: A polynomial-time nuclear vector replacement algorithm for automated NMR resonance assignments. In: *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pp. 176–187 (2003)
38. Langmead, C.J., Donald, B.R.: An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *J. Biomol. NMR* 29(2), 111–138 (2004)
39. Leach, A.R., Lemon, A.P.: Exploring the conformational space of protein side chains using dead-end elimination and the A\* algorithm. *Proteins* 33(2), 227–239 (1998)
40. Li, K.B., Sanctuary, B.C.: Automated extracting of amino acid spin systems in proteins using 3D HCCH-COSY/TOCSY spectroscopy and constrained partitioning algorithm (CPA). *J. Chem. Inf. Comput. Sci.* 36, 585–593 (1996)
41. Li, K.B., Sanctuary, B.C.: Automated resonance assignment of proteins using heteronuclear 3D NMR. 2. Side chain and sequence-specific assignment. *J. Chem. Inf. Comput. Sci.* 37, 467–477 (1997)
42. Li, M., Phatnani, H.P., Guan, Z., Sage, H., Greenleaf, A.L., Zhou, P.: Solution structure of the Set2-Rpb1 interacting domain of human Set2 and its interaction with the hyperphosphorylated C-terminal domain of Rpb1. *Proceedings of the National Academy of Sciences* 102, 17636–17641 (2005)
43. Lin, Y., Wagner, G.: Efficient side-chain and backbone assignment in large proteins: Application to tGCN5. *J. Biomol. NMR* 15, 227–239 (1999)
44. Linge, J.P., Habeck, M., Rieping, W., Nilges, M.: ARIA: Automated NOE assignment and NMR structure calculation. *Bioinformatics* 19(2), 315–316 (2003)
45. Looger, L.L., Hellinga, H.W.: Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J. Mol. Biol.* 300(1), 429–445 (2001)

46. Marin, A., Malliavin, T.E., Nicolas, P., Delsuc, M.A.: From NMR chemical shifts to amino acid types: investigation of the predictive power carried by nuclei. *Journal of Biomolecular NMR* 30, 47 (2004)
47. Masse, J.E., Keller, R., Pervushin, K.: SideLink: automated side-chain assignment of biopolymers from NMR data by relative-hypothesis-prioritization-based simulated logic. *Journal of Magnetic Resonance* 181(1), 45–67 (2006)
48. Montelione, G.T., Moseley, H.N.B.: Automated analysis of NMR assignments and structures for proteins. *Curr. Opin. Struct. Biol.* 9, 635–642 (1999)
49. Mumenthaler, C., Güntert, P., Braun, W., Wüthrich, K.: Automated combined assignment of NOESY spectra and three-dimensional protein structure determination. *Journal of Biomolecular NMR* 10(4), 351–362 (1997)
50. Ottiger, M., Delaglio, F., Bax, A.: Measurement of J and dipolar couplings from simplified two-dimensional NMR spectra. *Journal of Magnetic Resonance* 138, 373–378 (1998)
51. Prestegard, J.H., Bougault, C.M., Kishore, A.I.: Residual Dipolar Couplings in Structure Determination of Biomolecules. *Chemical Reviews* 104, 3519–3540 (2004)
52. Rieping, W., Habeck, M., Nilges, M.: Inferential Structure Determination. *Science* 309, 303–306 (2005)
53. Ruan, K., Briggman, K.B., Tolman, J.R.: De novo determination of internuclear vector orientations from residual dipolar couplings measured in three independent alignment media. *Journal of Biomolecular NMR* 41, 61–76 (2008)
54. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs (2002)
55. Schwieters, C.D., Kuszewski, J.J., Tjandra, N., Clore, G.M.: The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.* 160, 65–73 (2003)
56. Sun, X., Druzdzel, M.J., Yuan, C.: Dynamic Weighting A\* Search-Based MAP Algorithm for Bayesian Networks. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2385–2390 (2007)
57. Tjandra, N., Bax, A.: Direct measurement of distances and angles in biomolecules by NMR in a dilute liquid crystalline medium. *Science* 278, 1111–1114 (1997)
58. Tolman, J.R., Flanagan, J.M., Kennedy, M.A., Prestegard, J.H.: Nuclear magnetic dipole interactions in field-oriented proteins: Information for structure determination in solution. *Proc. Natl. Acad. Sci. USA* 92, 9279–9283 (1995)
59. Ulrich, E.L., Akutsu, H., Dorelejers, J.F., Harano, Y., Ioannidis, Y.E., Lin, J., Livny, M., Mading, S., Maziuk, D., Miller, Z., Nakatani, E., Schulte, C.F., Tolmie, D.E., Wenger, R.K., Yao, H., Markley, J.L.: BioMagResBank. *Nucleic Acids Research* 36, D402–D408 (2007)
60. Vitek, O., Bailey-Kellogg, C., Craig, B., Vitek, J.: Inferential backbone assignment for sparse data. *J. Biomolecular NMR* 35, 187–208 (2006)
61. Wang, L., Donald, B.R.: Exact solutions for internuclear vectors and backbone dihedral angles from NH residual dipolar couplings in two media, and their application in a systematic search algorithm for determining protein backbone structure. *Jour. Biomolecular NMR* 29(3), 223–242 (2004)
62. Wang, L., Mettu, R., Donald, B.R.: A Polynomial-Time Algorithm for De Novo Protein Backbone Structure Determination from NMR Data. *Journal of Computational Biology* 13(7), 1276–1288 (2006)
63. Wei, Z., Li, H.: A Markov random field model for network-based analysis of genomic data. *Bioinformatics* 23, 1537–1544 (2007)
64. Wu, K.-P., Chang, J.-M., Chen, J.-B., Chang, C.-F., Wu, W.-J., Huang, T.-H., Sung, T.-Y., Hsu, W.-L.: RIBRA-an Error-Tolerant Algorithm for the NMR Backbone Assignment Problem. In: *Proceedings of the International conference on Research in Computational Molecular Biology (RECOMB 2005)*, pp. 229–244 (2005)

65. Xu, Y., Xu, D., Uberbacher, E.C.: An efficient computational method for globally optimal threading. *J. Comput. Biol.* 5(3), 597–614 (1998)
66. Zeng, J., Boyles, J., Tripathy, C., Wang, L., Yan, A., Zhou, P., Donald, B.R.: High-Resolution Protein Structure Determination Starting with a Global Fold Calculated from Exact Solutions to the RDC Equations. *Journal of Biomolecular NMR* 45, 265–281 (2009)
67. Zeng, J., Zhou, P., Donald, B.R.: A Markov Random Field Framework for Protein Side-Chain Resonance Assignment – Supplementary Material. Department of Computer Science, Duke University (January 2010), <http://www.cs.duke.edu/donaldlab/Supplementary/recomb10/>
68. Zimmerman, D.E., Kulikowski, C.A., Feng, W., Tashiro, M., Chien, C.-Y., Ríos, C.B., Moy, F.J., Powers, R., Montelione, G.T.: Automated analysis of protein NMR assignments using methods from artificial intelligence. *J. Mol. Biol.* 269, 592–610 (1997)

# Genomic DNA $k$ -mer Spectra: Models and Modalities\*

Benny Chor<sup>1</sup>, David Horn<sup>1</sup>, Nick Goldman<sup>2</sup>,  
Yaron Levy<sup>1</sup>, and Tim Massingham<sup>2</sup>

<sup>1</sup> School of Computer Science, Tel Aviv University, Israel

<sup>2</sup> European Bioinformatics Institute, Hinxton, Cambridge, UK  
benny@cs.tau.ac.il, horn@tau.ac.il, goldman@ebi.ac.uk,  
yaron@emza-vs.com, tim.massingham@ebi.ac.uk

## Abstract

**Background:** The empirical frequencies of DNA  $k$ -mers in whole genome sequences provide an interesting perspective on genomic complexity, and the availability of large segments of genomic sequence from many organisms means that analysis of  $k$ -mers with non-trivial lengths is now possible.

**Results:** We have studied the  $k$ -mer spectra of more than 100 species from Archea, Bacteria, and Eukaryota, particularly looking at the modalities of the distributions. As expected, most species have a unimodal  $k$ -mer spectrum. However, a few species, including all mammals, have multimodal spectra. These species coincide with the tetrapods. Genomic sequences are clearly very complex, and cannot be fully explained by any simple probabilistic model. Yet we sought such an explanation for the observed modalities, and discovered that low-order Markov models capture this property (and some others) fairly well.

**Conclusions:** Multimodal spectra are characterized by specific ranges of values of C+G content and of CpG dinucleotide suppression, a range that encompasses all tetrapods analyzed. Other genomes, like that of the protozoa *Entamoeba histolytica*, which also exhibits CpG suppression, do not have multimodal  $k$ -mer spectra. Groupings of functional elements of the human genome also have a clear modality, and exhibit either a unimodal or multimodal behaviour, depending on the two above mentioned values.

**Keywords:** DNA  $k$ -mers, whole genome empirical frequencies, low-order Markov models, modalities of DNA distributions, phylogenetic characterization of modalities.

---

\* Complete version can be found at <http://genomebiology.com/2009/10/10/R108>.

# Deciphering the Swine-Flu Pandemics of 1918 and 2009

Richard Goldstein, Mario dos Reis, Asif Tamuri, and Alan Hay

NIMR, United Kingdom

richard.goldstein@nimr.mrc.ac.uk, mdosrei@nimr.mrc.ac.uk,  
atamuri@nimr.mrc.ac.uk, ahay@nimr.mrc.ac.uk

**Abstract.** The devastating “Spanish flu” of 1918 killed an estimated 50 million people worldwide, ranking it as the deadliest pandemic in recorded human history. It is generally believed that the virus transferred from birds directly to humans shortly before the start of the pandemic, subsequently jumping from humans to swine. By developing ‘non-homogeneous’ substitution models that consider that substitution patterns may be different in human, avian, and swine hosts, we can determine the timing of the host shift to mammals. We find it likely that the Spanish flu of 1918, like the current 2009 pandemic, was a ‘swine-origin’ influenza virus. Now that we are faced with a new pandemic, can we understand how influenza is able to change hosts? Again by modelling the evolutionary process, considering the different selective constraints for viruses in the different hosts, we can identify locations that seem to be under different selective constraints in humans and avian hosts. This allows us to identify changes that may have facilitated the establishment of the 2009 swine-origin flu in humans.

The swine-origin pandemic of 2009 highlighted the importance of understanding the process of host shifts in zoonotic pathogens. Understanding past host shifts can provide important information about where current threats may originate. Analysing past host shifts using molecular evolutionary analysis has been limited by models of sequence change that consider the substitution process to be the same for all locations at all time (at most modulated by a site-specific scaling factor). By considering non-homogeneous non-stationary models, we can model how the substitution process in viruses such as influenza differs at different locations in the proteins and in different hosts. This allows us to determine the timing and trajectory of host shift events, as well as identify locations where changes in amino acid may assist or be required for the host shift event to occur. We apply these methods to the 1918 ‘Spanish Flu’ pandemic, determining that, contrary to what is widely believed, this pandemic also was likely from a swine-origin virus. We then apply these techniques to the 2009 influenza pandemic, determining locations where changes in amino acid may have facilitated the ability of these viruses to sift from swine to human hosts. These models have wide applicability where changes in selective constraints might have occurred, including understanding other pathogen host shifts, deciphering how HIV responds to drug treatment, and how to understand and predict changes in functionality or physiological context of proteins.

## References

1. dos Reis, M., Hay, A.J., Goldstein, R.A.: Using Non-Homogeneous Models of Nucleotide Substitution to Identify Host Shift Events: Application to the Origin of the 1918 'Spanish' Influenza Pandemic Virus. *J. Mol. Evol.* 69, 333–345 (2009)
2. Tamuri, A.U., dos Reis, M., Hay, A.J., Goldstein, R.A.: Identifying changes in selective constraints: host shifts in influenza. *PLoS Comput. Biol.* 5, 31000564 (2009)

# Distinguishing Direct versus Indirect Transcription Factor-DNA Interactions

Raluca Gordân<sup>1</sup>, Alexander J. Hartemink<sup>2</sup>, and Martha L. Bulyk<sup>1,3,4</sup>

<sup>1</sup> Division of Genetics, Dept. of Medicine, Brigham & Women's Hospital and Harvard Medical School, Boston, MA 02115, USA

<sup>2</sup> Duke University, Dept. of Computer Science, Box 90129, Durham, NC 27708, USA

<sup>3</sup> Dept. of Pathology, Brigham & Women's Hospital and Harvard Medical School, Boston, MA 02115, USA

<sup>4</sup> Harvard-MIT Division of Health Sciences and Technology (HST), Harvard Medical School, Boston, MA 02115, USA

rgordan@rics.bwh.harvard.edu,  
amink@cs.duke.edu,  
mlbulyk@receptor.med.harvard.edu

**Abstract.** Transcriptional regulation is largely enacted by transcription factors (TFs) binding DNA. Large numbers of TF binding motifs have been revealed by ChIP-chip experiments followed by computational DNA motif discovery. However, the success of motif discovery algorithms has been limited when applied to sequences bound *in vivo* (such as those identified by ChIP-chip) because the observed TF-DNA interactions are not necessarily direct: some TFs predominantly associate with DNA indirectly through protein partners, while others exhibit both direct and indirect binding.

We present the first method for distinguishing between direct and indirect TF-DNA interactions, integrating *in vivo* TF binding data, *in vivo* nucleosome occupancy data, and DNA binding motifs from *in vitro* protein binding microarray experiments. When applied to yeast ChIP-chip data, our method reveals that only 48% of the data sets can be readily explained by direct binding of the profiled TF, while 16% can be explained by indirect DNA binding. In the remaining 36%, none of the motifs used in our analysis was able to explain the ChIP-chip data, either because the data were too noisy or because the set of motifs was incomplete. As more *in vitro* TF DNA binding motifs become available, our method could be used to build a complete catalog of direct and indirect TF-DNA interactions. Our method is not restricted to yeast or to ChIP-chip data, but can be applied in any system for which both *in vivo* binding data and *in vitro* DNA binding motifs are available.



# A Self-regulatory System of Interlinked Signaling Feedback Loops Controls Mouse Limb Patterning

Jean-Denis Benazet<sup>1</sup>, Mirko Bischofberger<sup>2</sup>, Eva Tiecke<sup>1</sup>,  
Alexandre Goncalves<sup>1</sup>, James F. Martin<sup>3</sup>, Aime Zuniga<sup>1</sup>,  
Felix Naef<sup>2</sup>, and Rolf Zeller<sup>1</sup>

<sup>1</sup> Developmental Genetics, Department of Biomedicine,  
University of Basel, Switzerland

<sup>2</sup> Computational Systems Biology Group,  
Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

<sup>3</sup> Texas A&M Health Science Center,  
Institute of Biosciences and Technology, Houston, USA

**Abstract.** Developmental pathways need to be robust against environmental and genetic variation to enable reliable morphogenesis. Here, we take a systems biology approach to explain how robustness is achieved in the developing mouse limb, a classical model of organogenesis. By combining quantitative genetics with computational modeling we established a computational model of multiple interlocked feedback modules, involving sonic hedgehog (SHH) morphogen, fibroblast growth factor (FGFs) signaling, bone morphogenetic protein (BMP) and its antagonist GREM1. Earlier modeling work had emphasized the versatile kinetic characteristics of interlocked feedback loops operating at different time scales. Here we develop and then validate a similar computational model to show how BMP4 first initiates and SHH then propagates feedback in the network through differential transcriptional regulation of *Grem1* to control digit specification. This switch occurs by linking a fast BMP4/GREM1 module to a slower SHH/GREM1/FGF feedback loop. Simulated gene expression profiles modeled normal limb development as well those of single-gene knockouts. Sensitivity analysis showed how the model was robust and insensitive to variability in parameters. A surprising prediction of the model was that an early *Bmp4* signal is essential to kick-start *Grem1* expression and the digit specification system. We experimentally validated the prediction using inducible alleles and showed that early, but not late, removal of *Bmp4* dramatically disrupted limb development. Sensitivity analysis showed how robustness emerges from this circuitry. This study shows how modeling and computation can help us understand how self-regulatory signaling networks achieve robust regulation of limb development, by exploiting interconnectivity among the three signaling pathways. We expect that similar computational analyses will shed light on the origins of robustness in other developmental systems, and I will discuss some recent examples from our ongoing research on developmental patterning.

## Reference

1. Benazet, J.D., Bischofberger, M., Tiecke, E., Goncalves, A., Martin, J.F., Zuniga, A., Naef, F., Zeller, R.: A Self-Regulatory System of Interlinked Signaling Feedback Loops Controls Mouse Limb Patterning. *Science* 323(5917), 1050–1053 (2009)

# Automated High-Dimensional Flow Cytometric Data Analysis

Saunmyadipta Pyne<sup>1</sup>, Xinli Hu<sup>1</sup>, Kui Wang<sup>2</sup>, Elizabeth Rossin<sup>1</sup>, Tsung-I Lin<sup>3</sup>,  
Lisa Maier<sup>1</sup>, Clare Baecher-Allan<sup>4</sup>, Geoffrey McLachlan<sup>2</sup>, Pablo Tamayo<sup>1</sup>,  
David Hafler<sup>1</sup>, Philip De Jager<sup>1</sup>, and Jill Mesirov<sup>1</sup>

<sup>1</sup> Broad Institute of MIT and Harvard, United States

<sup>2</sup> University of Queensland, Australia

<sup>3</sup> Department of Applied Mathematics, National Chung Hsing University, Taiwan

<sup>4</sup> Division of Molecular Immunology, Center for Neurologic Diseases,  
Brigham and Women's Hospital and Harvard Medical School, United States

**Abstract.** Flow cytometry is widely used for single cell interrogation of surface and intracellular protein expression by measuring fluorescence intensity of fluorophore-conjugated reagents. We focus on the recently developed procedure of Pyne et al. (2009, Proceedings of the National Academy of Sciences USA 106, 8519-8524) for automated high-dimensional flow cytometric analysis called FLAME (FLow analysis with Automated Multivariate Estimation). It introduced novel finite mixture models of heavy-tailed and asymmetric distributions to identify and model cell populations in a flow cytometric sample. This approach robustly addresses the complexities of flow data without the need for transformation or projection to lower dimensions. It also addresses the critical task of matching cell populations across samples that enables downstream analysis. It thus facilitates application of flow cytometry to new biological and clinical problems. To facilitate pipelining with standard bioinformatic applications such as high-dimensional visualization, subject classification or outcome prediction, FLAME has been incorporated with the GenePattern package of the Broad Institute. Thereby analysis of flow data can be approached similarly as other genomic platforms. We also consider some new work that proposes a rigorous and robust solution to the registration problem by a multi-level approach that allows us to model and register cell populations simultaneously across a cohort of high-dimensional flow samples. This new approach is called JCM (Joint Clustering and Matching). It enables direct and rigorous comparisons across different time points or phenotypes in a complex biological study as well as for classification of new patient samples in a more clinical setting.

# Discovering Transcriptional Modules by Combined Analysis of Expression Profiles and Regulatory Sequences\*

Yonit Halperin\*\*, Chaim Linhart\*\*, Igor Ulitsky, and Ron Shamir

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel  
{yonithal, chaiml, ulitskyi, rshamir}@tau.ac.il

**Abstract.** A key goal of gene expression analysis is the characterization of transcription factors (TFs) and micro-RNAs (miRNAs) regulating specific transcriptional programs. The most common approach to address this task is a two-step methodology: In the first step, a clustering procedure is executed to partition the genes into groups that are believed to be co-regulated, based on expression profile similarity. In the second step, a motif discovery tool is applied to search for over-represented cis-regulatory motifs within each group. In an effort to obtain better results by simultaneously utilizing all available information, several studies have suggested computational schemes for a single-step combined analysis of expression and sequence data. Despite extensive research, reverse engineering complex regulatory networks from microarray measurements remains a difficult challenge with limited success, especially in metazoans.

We present Allegro [1], a new method for de-novo discovery of TF and miRNA binding sites through joint analysis of genome-wide expression data and promoter or 3' UTR sequences. In brief, Allegro enumerates a huge number of candidate motifs in a series of refinement phases to converge to high-scoring motifs. For each candidate motif, it executes a cross-validation-like procedure to learn an expression model that describes the shared expression profile of the genes, whose cis-regulatory sequence contains the motif. It then computes a p-value for the over-representation of the motif within the genes that best fit the expression profile. The output of Allegro is a non-redundant list of top-scoring motifs and the expression patterns they induce.

The expression model used by Allegro is a novel log likelihood-based, non-parametric model, analogous to the position weight matrix commonly used for representing TF binding sites. Unlike most extant methods, our approach does not assume that the expression values follow a pre-defined type of distribution, and can capture transcriptional modules whose expression profiles differ from the rest of the genome across a small fraction of the conditions. Furthermore, it successfully handles cases where the expression levels are correlated to the length and GC-content of the cis-regulatory sequences. Such correlations are quite common in practice, and often bias existing techniques, leading to false predictions and low sensitivity.

---

\* Supported in part by the Israel Science Foundation (grant 802/08 and Converging Technologies Program grant 1767.07).

\*\* These authors contributed equally to this work.

Allegro introduces several additional unique ideas and features, and is implemented in a graphical, user-friendly software tool. We apply it on several large datasets (>100 conditions), in murine, fly and human, report on the transcriptional modules it uncovers, and show that it outperforms extant techniques. Allegro is available at <http://acgt.cs.tau.ac.il/allegro>.

## Reference

1. Halperin, et al.: *Nucleic Acids Research* 37(5), 1566–1579 (2009)

# Author Index

- Aebersold, Ruedi 96  
Agarwala, Sudeep 110  
Aguiar, Derek 158  
Altun, Yasemin 522  
Atias, Nir 1  
Ay, Ferhat 15
- Backofen, Rolf 473  
Baecher-Allan, Clare 577  
Bafna, Vineet 310  
Bandeira, Nuno 208  
Bedo, Justin 297  
Benazet, Jean-Denis 575  
Bercovici, Sivan 31, 50  
Beresford-Smith, Bryan 297  
Bilmes, Jeff A. 441  
Bischofberger, Mirko 575  
Blelloch, Guy 369  
Brudno, Michael 357  
Buhmann, Joachim M. 96  
Bulyk, Martha L. 574
- Caldas, José 65  
Chance, Mark R. 80  
Chin, Francis Y.L. 426  
Chor, Benny 571  
Chowdhury, Salim A. 80  
Claassen, Manfred 96  
Conway, Thomas 297
- Danford, Timothy 110  
Daskalakis, Constantinos 123  
De Jager, Philip 577  
Diekhans, Mark 410  
Donald, Bruce Randall 550  
dos Reis, Mario 572  
Dowell, Robin 110
- Earl, Dent 410
- Feng, Jianxing 138  
Fink, Gerald 110
- Gao, Xin 189  
Geiger, Dan 31
- Gifford, David 110  
Girdea, Marta 384  
Goldman, Nick 571  
Goldstein, Richard 572  
Gonalves, Alexandre 575  
Gordân, Raluca 574  
Grisafi, Paula 110  
Gusev, Alexander 491
- Habeck, Michael 174  
Hafler, David 577  
Halldórsson, Bjarni V. 158  
Halperin, Eran 397  
Halperin, Yonit 578  
Hartemink, Alexander J. 574  
Haussler, David 410  
Hay, Alan 572  
Hirsch, Michael 174  
Hoch, Allison 456  
Horn, David 571  
Hsu, David 281  
Hu, Xinli 577
- Istrail, Sorin 158
- Jang, Richard 189  
Jeong, Kyowon 208  
Jiang, Tao 138  
Jojic, Vladimir 341
- Kahveci, Tamer 15  
Kao, Wei-Chun 233  
Kaski, Samuel 65  
Kelley, David R. 248  
Khuller, Samir 456  
Kim, Sangtae 208  
Kim, Yoo-Ah 263  
Kingsford, Carl 248  
Koh, Geoffrey 281  
Koller, Daphne 341  
Kowalczyk, Adam 297  
Koyutürk, Mehmet 80  
Kozanitis, Christos 310  
Kruglyak, Semyon 310  
Kucherov, Gregory 384

- Lam, Fumei 325  
 Langley, Charles H. 325  
 Laserson, Jonathan 341  
 Lee, Seunghak 357  
 Leiva, Jose 522  
 Leung, Henry C.M. 426  
 Levy, Yaron 571  
 Li, Ming 189  
 Li, Wei 138  
 Lin, Tsung-I 577  
 Linhart, Chaim 578
- Ma, Jian 410  
 Maier, Lisa 577  
 Martin, James F. 575  
 Massingham, Tim 571  
 McLachlan, Geoffrey 577  
 Mesirov, Jill 577  
 Misra, Navodit 369  
 Möhl, Mathias 473
- Naef, Felix 575  
 Nibbe, Rod K. 80  
 Noble, William Stafford 441  
 Noé, Laurent 384
- Paşaniuc, Bogdan 397  
 Paten, Benedict 410  
 Pe'er, Itsik 491  
 Peng, Yu 426  
 Pevzner, Pavel A. 208  
 Pinter, Ron Y. 50  
 Przytycka, Teresa M. 263  
 Pyne, Saumyadipta 577
- Raphael, Benjamin J. 506  
 Raschid, Louiqa 456  
 Rättsch, Gunnar 522  
 Ravi, R. 369  
 Reynolds, Sheila M. 441  
 Roch, Sebastien 123  
 Rossin, Elizabeth 577
- Saha, Barna 456  
 Sahinalp, S. Cenk 473  
 Salari, Raheleh 473  
 Saunders, Chris 310  
 Schölkopf, Bernhard 174  
 Schwartz, Russell 369  
 Setty, Manu N. 491  
 Shamir, Ron 578  
 Sharan, Roded 1  
 Sharon, Itai 50  
 Shlomi, Tomer 50  
 Song, Yun S. 233, 325  
 St. John, John 410  
 Suh, Bernard 410
- Tamayo, Pablo 577  
 Tamuri, Asif 572  
 Tarpine, Ryan 158  
 Thiagarajan, P.S. 281  
 Tietze, Eva 575
- Ulitsky, Igor 578  
 Upfal, Eli 506
- Vandin, Fabio 506  
 Varghese, George 310
- Wang, Kui 577  
 Weng, Zhiping 441  
 Widmer, Christian 522  
 Will, Sebastian 473  
 Wu, Yu-Wei 535  
 Wuchty, Stefan 263
- Xing, Eric 357
- Ye, Yuzhen 535  
 Yiu, S.M. 426
- Zaitlen, Noah 397  
 Zeller, Rolf 575  
 Zeng, Jianyang 550  
 Zhang, Xiao-Ning 456  
 Zhou, Pei 550  
 Zuniga, Aime 575