

Benjamin J. Raphael (Ed.)

LNBI 10812

Research in Computational Molecular Biology

22nd Annual International Conference, RECOMB 2018
Paris, France, April 21–24, 2018
Proceedings

 Springer

Subseries of Lecture Notes in Computer Science

LNBI 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

LNBI Editorial Board

Søren Brunak

Technical University of Denmark, Kongens Lyngby, Denmark

Mikhail S. Gelfand

IITP, Research and Training Center on Bioinformatics, Moscow, Russia

Thomas Lengauer

Max Planck Institute for Informatics, Saarbrücken, Germany

Satoru Miyano

University of Tokyo, Tokyo, Japan

Eugene Myers

Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

Marie-France Sagot

Université Lyon 1, Villeurbanne, France

David Sankoff

University of Ottawa, Ottawa, Canada

Ron Shamir

Tel Aviv University, Ramat Aviv, Tel Aviv, Israel

Terry Speed

Walter and Eliza Hall Institute of Medical Research, Melbourne, VIC, Australia

Martin Vingron

Max Planck Institute for Molecular Genetics, Berlin, Germany

W. Eric Wong

University of Texas at Dallas, Richardson, TX, USA

More information about this series at <http://www.springer.com/series/5381>

Benjamin J. Raphael (Ed.)

Research in Computational Molecular Biology

22nd Annual International Conference, RECOMB 2018
Paris, France, April 21–24, 2018
Proceedings

Editor

Benjamin J. Raphael
Computer Science Department
Princeton University
Princeton, NJ
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Bioinformatics
ISBN 978-3-319-89928-2 ISBN 978-3-319-89929-9 (eBook)
<https://doi.org/10.1007/978-3-319-89929-9>

Library of Congress Control Number: 2018940142

LNCS Sublibrary: SL8 – Bioinformatics

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains 38 extended and short abstracts presented at the 22nd International Conference on Research in Computational Molecular Biology (RECOMB) 2018, which was hosted by the Pierre et Marie Curie Campus of Sorbonne Université in Paris, April 21–24.

These 38 contributions were selected from a total of 193 submissions. Each submission was reviewed by three members of the Program Committee (PC), who in many cases solicited additional advice from external reviewers. Following the initial reviews, final decisions were made after an extensive discussion of the submissions among the members of the PC. Reviews and discussions were conducted through the EasyChair Conference Management System.

While RECOMB 2018 did not allow parallel submissions, authors of accepted papers were given the option to publish short abstracts in these proceedings and submit their full papers to a journal. In addition, several accepted papers were invited to submit revised manuscripts for consideration in *Cell Systems*. Papers accepted for oral presentation that were subsequently submitted to a journal are published as short abstracts and were deposited on the preprint server arxiv.org or biorxiv.org. All other papers that appear as long abstracts in the proceedings were invited for submission to the RECOMB 2018 special issue of the *Journal of Computational Biology*.

In addition to presentations of these contributed papers, RECOMB 2018 also featured six invited keynote talks given by leading scientists. The keynote speakers were Peter Campbell (Sanger Institute), Nevan Krogan (University of California, San Francisco), Ron Shamir (Tel Aviv University), François Spitz (Institut Pasteur), Sarah Teichmann (Sanger Institute), and Tandy Warnow (University of Illinois, Urbana Champaign).

RECOMB 2018 also featured highlight talks of computational biology papers that were published in journals during the previous 18 months. Of the 27 highlight submissions, seven were selected for oral presentation at RECOMB.

The success of RECOMB depends on dedicated efforts and a substantial investment of time from many colleagues. I especially thank the co-chairs of the Organizing Committee, Mireille Régnier (Ecole Polytechnique) and Yann Ponty (CNRS, Ecole Polytechnique), for hosting RECOMB; the Steering Committee and especially its chair, Bonnie Berger (MIT), for help, advice, and support throughout the process; Cenk Sahinalp (Indiana University), the Program Chair of RECOMB 2017, for answering my many questions; Ziv Bar Joseph (CMU) for chairing the highlights track; Christina Boucher (University of Florida) for chairing the poster track; Mohammed El-Kebir (UIUC) for serving as the publications chair; Alessandra Carbone (Sorbonne Université) for acting as the publicity chair; H el ene Touzet (CNRS, Universit e Lille I) for organizing the RECOMB Satellites; Mark Chaisson (University of Southern California), Rayan Chikhi (CNRS, University of Lille, France), Valentina Boeva (Institut Cochin and Inserm), Moritz Gerstung (EMBL-EBI), Hugues Aschard (Institut

Pasteur), Simon Gravel (McGill University), Olivier de Fresnoye, Julio Saez-Rodriguez, Pablo Meyer-Rojas, Gustavo Stolovitzky, and Elise Blaese for chairing the RECOMB Satellite Workshops on Massively Parallel Sequencing, Computational Cancer Biology, Genetics, and DREAM Challenges.

I also thank the PC members and external reviewers for their timely reviews of assigned papers despite their busy schedules; the authors of the papers, highlights, and posters for their scientific contributions; and all the attendees for their enthusiastic participation in the conference.

Finally, I also thank our sponsors, including the International Society of Computational Biology (ISCB), who supported student travel fellowships, as well as the Centre National de la Recherche Scientifique (CNRS), IBM Research, Ecole Polytechnique, Réseau Francilien en Sciences Informatiques (RFSI), Région Ile de France, Frontiers, PRABI, Inria, and GDR BIM, and Sorbonne Université.

February 2018

Ben Raphael

Organization

Program Committee

Max Alekseyev	George Washington University, USA
Rolf Backofen	Albert-Ludwigs-University Freiburg, Germany
Vineet Bafna	University of California San Diego, USA
Ziv Bar-Joseph	Carnegie Mellon University, USA
Bonnie Berger	Massachusetts Institute of Technology, USA
Mathieu Blanchette	McGill University, Canada
Lenore Cowen	Tufts University, USA
Raluca Gordan	Duke University, USA
Fereydoun Hormozdiari	University of Washington, USA
John Kececioгу	University of Arizona, USA
Carl Kingsford	Carnegie Mellon University, USA
Gunnar W. Klau	Heinrich Heine University of Düsseldorf, Germany
Jens Lagergren	SBC and CSC, KTH, Sweden
Mark Leiserson	University of Maryland, USA
Ming Li	University of Waterloo, Canada
Veli Mäkinen	University of Helsinki, Finland
Jian Ma	Carnegie Mellon University, USA
Paul Medvedev	The Pennsylvania State University, USA
Bernard Moret	Ecole Polytechnique Fédérale de Lausanne, Switzerland
Layla Oesper	Carleton College, USA
Laxmi Parida	IBM, USA
Itsik Pe'Er	Columbia University, USA
Jian Peng	University of Illinois at Urbana-Champaign, USA
Yann Ponty	CNRS/LIX, Polytechnique, France
Teresa Przytycka	National Institutes of Health, USA
Ben Raphael	Princeton University, USA
Mireille Regnier	Inria, France
Knut Reinert	FU Berlin, Germany
S. Cenk Sahinalp	Indiana University Bloomington, USA
Michael Schatz	Cold Spring Harbor Laboratory, USA
Alexander Schoenhuth	Vrije Universiteit Amsterdam, The Netherlands
Russell Schwartz	Carnegie Mellon University, USA
Roded Sharan	Tel Aviv University, Israel
Mona Singh	Princeton University, USA
Donna Slonim	Tufts University, USA
Sagi Snir	Institute of Evolution

Jens Stoye	Bielefeld University, Germany
Fengzhu Sun	University of Southern California, USA
Wing-Kin Sung	National University of Singapore
Ewa Szczurek	University of Warsaw, Poland
Glenn Tesler	University of California San Diego, USA
Tamir Tuller	Tel Aviv University, Israel
Alfonso Valencia	Barcelona Supercomputing Center, Spain
Fabio Vandin	University of Padova, Italy
Jean-Philippe Vert	Ecole des Mines de Paris, France
Martin Vingron	Max Planck Institut für molekulare Genetik, Germany
Jerome Waldispuhl	McGill University, Canada
Tandy Warnow	University of Illinois at Urbana-Champaign, USA
Sebastian Will	University of Vienna, Austria
Jinbo Xu	Toyota Technological Institute at Chicago, USA
Noah Zaitlen	University of California San Francisco, USA
Alex Zelikovsky	Georgia State University, USA
Jianyang Zeng	Tsinghua University, China
Louxin Zhang	National University of Singapore
Michal Ziv-Ukelson	Ben-Gurion University of the Negev, Israel

Additional Reviewers

Adam, Alex	Bosio, Mattia	Dojer, Norbert
Aganezov, Sergey	Bruner, Ariel	Dunin-Horkawicz, Stanislaw
Akbari, Ali	Bryant, David	Durham, Timothy
Alavi, Amir	Burch, Kathryn	Eaton, Jesse
Alexeev, Nikita	Canzar, Stefan	Engler, Martin
Almodaresi, Fatemeh	Casale, Paolo	Eyras, Eduardo
Amodio, Matthew	Chaisson, Mark	Fan, Jason
Avdeyev, Pavel	Chang, Kingsley	Fang, Han
Aviran, Sharon	Chhangawala, Sagar	Feinberg, Adam
Avni, Eliran	Chor, Benny	Feizi, Soheil
Bakhtiari, Mehrdad	Ciccolella, Simone	Fernandez-Recio, Juan
Balvert, Marleen	Cirillo, Davide	Fleischauer, Markus
Bankevich, Anton	Costa, Fabrizio	Ford, Michael
Basu, Aritra	Cui, Xuefeng	Frenkel, Vladimir
Batut, Bérénice	Dadi, Temesgen	Frishberg, Amit
Beissbarth, Tim	Dahl, Andy	Gaudelet, Thomas
Benner, Philipp	Danko, David	Gawronski, Alexander
Bepler, Tristan	Davila, Jose	Gawrychowski, Pawel
Bernstein, Ryan	Deblasio, Dan	Ghareghani, Maryam
Bilenne, Olivier	Dilthey, Alexander	Giambartolomei, Claudia
Biran, Hadas	Ding, Jun	Gigante, Scott
Bockmayr, Alexander	Dirmeier, Simon	Gogolewski, Krzysztof
Boix, Carles	Doerr, Daniel	

Guo, Yuchun	Lapierre, Nathan	Neubert, Kerstin
Gupta, Chhedi	Le, Thong	Ng, Bernard
Haghshenas, Ehsan	Lee, Heewook	Nguyen, Duong
Hammelmann, Jen	Lei, Haoyun	Noutahi, Emmanuel
Harel, Tom	Levi, Maya	Ntranos, Vasilis
Harrison, Robert	Li, Han	Numanagic, Ibrahim
He, Liang	Li, Shuai Cheng	Nurk, Sergey
Heinig, Matthias	Lin, Chieh	Ochoa, Alex
Heinrich, Verena	Lin, Dejun	Ochoa, Idoia
Herman-Iżycka, Julia	Liptak, Zsuzsanna	Oren, Yael
Hescott, Benjamin	Liu, Ge	Osmanbeyoglu, Hatice
Hodzic, Ermin	Liu, Jie	Ouangraoua, Aida
Hormozdiari, Farhad	Liu, Yaping	Pal, Soumitra
Hou, Lei	Lu, Yang	Park, Jisoo
Hristov, Borislav	Ludwig, Marcus	Patkar, Sushant
Hu, Alex	Luo, Yunan	Payne, Samuel
Hu, Yangyang	Ma, Cong	Persikov, Anton
Huska, Matt	Malikic, Salem	Peterlongo, Pierre
Huynh, Linh	Malod-Dognin, Noel	Pietras, Christopher
Iranmehr, Arya	Mancuso, Nicholas	Pirkli, Martin
Jahn, Katharina	Mandric, Igor	Pitkänen, Esa
Jain, Siddhartha	Mangul, Serghei	Pockrandt, Christopher
Jankowski, Aleksander	Mann, Martin	Posada-Céspedes, Susana
Jia, Peilin	Marass, Francesco	Preissner, Robert
Johnson, Ruth	Markowetz, Florian	Pritykin, Yuri
Joseph, Tyler	Marschall, Tobias	Pullman, Benjamin
Ju, Chelsea	Marz, Manja	Qu, Fangfang
Jünemann, Sebastian	Marçais, Guillaume	Raguideau, Sébastien
Kamath, Govinda	Maticzka, Daniel	Rahn, René
Keasar, Chen	May, Damon	Rajaby, Ramesh
Khabirova, Eleonora	Mazrouee, Sepideh	Rajaraman, Ashok
Kichaev, Gleb	Mefford, Joel	Ramisch, Anna
Kim, Jongkyu	Meleshko, Dmitrii	Rappoport, Nadav
Kim, Yoo-Ah	Mezlini, Aziz	Rashid, Sabrina
Kim, Younhun	Miladi, Milad	Rashidi Mehrabadi, Farid
Knijnenburg, Theo	Minkin, Ilia	Rautiainen, Mikko
Knyazev, Sergey	Momo, Remi	Razaviyayn, Meisam
Kockan, Can	Moon, Kevin	Reinharz, Vladimir
Komusiewicz, Christian	Moretti, Antonio	Ren, Jie
Koptagel, Hazal	Mueller, Jonas	Reyna, Matthew
Koslicki, David	Mueller, Teresa	Rhrissorkrai, Kahn
Kuipers, Jack	Müller, Tobias	Ricketts, Camir
Käll, Lukas	Na, Seungjin	Robinson, Welles
Köster, Johannes	Narasimhan, Giri	Rotman, Jeremy
Laehnemann, David	Naseri, Ardalan	Roytman, Megan
Lafond, Manuel	Navarro, Gonzalo	Ruffalo, Matthew

Russell, Nate	Starikovskaya, Tatiana	Wittler, Roland
Saez-Rodriguez, Julio	Sternberg, Barak	Wolf, Guy
Sahin, Merve	Steuerman, Yael	Wu, Guodong
Sahlin, Kristoffer	Strzalkowski, Alexander	Xia, Li
Salmela, Leena	Sun, Chen	Yang, Shuo
Sarmeshgi, Shahbeddin	Sundermann, Linda	Yankovitz, Gal
Sason, Itay	Swenson, Krister	Ye, Tiantian
Sauerwald, Natalie	Tang, Kujin	Yeo, Grace
Schmidt, Florian	Tejero, Héctor	Yu, Yun William
Schreiber, Jacob	Thomas, Marcus	Yuan, Han
Schrinner, Sven	Tran, Hieu Ngoc	Yuan, Jie
Schulte-Sasse, Roman	Tremblay-Savard, Olivier	Zamparo, Lee
Schulz, Tizian	Tsur, Dekel	Zehavi, Meirav
Sevillya, Gur	Tsyvina, Viachaslau	Zeng, Haoyang
Shao, Mingfu	van Dijk, David	Zhang, Jesse
Sheng, Wang	van Iersel, Leo	Zhang, Ruochi
Shi, Alvin	Veksler-Lublinsky, Isana	Zhang, Sai
Shi, Huwenbo	Vyatkina, Kira	Zhang, Zhizhuo
Shomorony, Ilan	Waldherr, Steffen	Zhong, Yi
Shrestha, Raunak	Wang, Hao	Zhou, Jian
Shteyman, Alan	Wang, Sheng	Zhu, Kaiyuan
Siegel, David	Wang, Weili	Zhu, Zifan
Silverbush, Dana	Wang, Yijie	
Skums, Pavel	Wang, Ying	
Solomon, Brad	Weimann, Oren	
Spang, Rainer	Wilenzik, Roni	
Srinivasan, Krishnan	Windels, Sam	
Standage, Daniel	Winkler, Jörg	

Contents

Long Reads Enable Accurate Estimates of Complexity of Metagenomes	1
<i>Anton Bankevich and Pavel Pevzner</i>	
Chromatyping: Reconstructing Nucleosome Profiles from NOME Sequencing Data	21
<i>Shounak Chakraborty, Stefan Canzar, Tobias Marschall, and Marcel H. Schulz</i>	
GTED: Graph Traversal Edit Distance	37
<i>Ali Ebrahimpour Boroojeny, Akash Shrestha, Ali Sharifi-Zarchi, Suzanne Renick Gallagher, S. Cenk Sahinalp, and Hamidreza Chitsaz</i>	
Statistical Inference of Peroxisome Dynamics	54
<i>Cyril Galitzine, Pierre M. Jean Beltran, Ileana M. Cristea, and Olga Vitek</i>	
Loss-Function Learning for Digital Tissue Deconvolution	75
<i>Franziska Görtler, Stefan Solbrig, Tilo Wettig, Peter J. Oefner, Rainer Spang, and Michael Altenbuchinger</i>	
Inference of Population Structure from Ancient DNA	90
<i>Tyler A. Joseph and Itsik Pe'er</i>	
Using Minimum Path Cover to Boost Dynamic Programming on DAGs: Co-linear Chaining Extended	105
<i>Anna Kuosmanen, Topi Paavilainen, Travis Gagie, Rayan Chikhi, Alexandru Tomescu, and Veli Mäkinen</i>	
Modeling Dependence in Evolutionary Inference for Proteins	122
<i>Gary Larson, Jeffrey L. Thorne, and Scott Schmidler</i>	
Constrained <i>De Novo</i> Sequencing of neo-Epitope Peptides Using Tandem Mass Spectrometry	138
<i>Sujun Li, Alex DeCourcy, and Haixu Tang</i>	
Reverse de Bruijn: Utilizing Reverse Peptide Synthesis to Cover All Amino Acid <i>k</i> -mers	154
<i>Yaron Orenstein</i>	
Circular Networks from Distorted Metrics	167
<i>Sebastien Roch and Kun-Chieh Wang</i>	

A Nested 2-Level Cross-Validation Ensemble Learning Pipeline Suggests a Negative Pressure Against Crosstalk snoRNA-mRNA Interactions in *Saccharomyces Cerevisiae*. 177
Antoine Soulé, Jean-Marc Steyaert, and Jérôme Waldispühl

Context-Specific Nested Effects Models. 194
Yuriy Sverchkov, Yi-Hsuan Ho, Audrey Gasch, and Mark Craven

Algorithmic Framework for Approximate Matching Under Bounded Edits with Applications to Sequence Analysis. 211
Sharma V. Thankachan, Chaitanya Aluru, Sriram P. Chockalingam, and Srinivas Aluru

Accurate Reconstruction of Microbial Strains from Metagenomic Sequencing Using Representative Reference Genomes. 225
Zhemín Zhou, Nina Luhmann, Nabil-Fareed Alikhan, Christopher Quince, and Mark Achtman

Short Papers

Targeted Genotyping of Variable Number Tandem Repeats with AdVNTR 243
Mehrdad Bakhtiari, Sharona Shleizer-Burko, Melissa Gymrek, Vikas Bansal, and Vineet Bafna

Positive-Unlabeled Convolutional Neural Networks for Particle Picking in Cryo-electron Micrographs. 245
Tristan Beppler, Andrew Morin, Alex J. Noble, Julia Brasch, Lawrence Shapiro, and Bonnie Berger

Designing RNA Secondary Structures Is Hard 248
Édouard Bonnet, Paweł Rzqżewski, and Florian Sikora

Generalizable Visualization of Mega-Scale Single-Cell Data. 251
Hyunghoon Cho, Bonnie Berger, and Jian Peng

Probabilistic Count Matrix Factorization for Single Cell Expression Data Analysis. 254
G. Durif, L. Modolo, J. E. Mold, S. Lambert-Lacroix, and F. Picard

Fixed-Parameter Tractable Sampling for RNA Design with Multiple Target Structures. 256
Stefan Hammer, Yann Ponty, Wei Wang, and Sebastian Will

Contribution of Structural Variation to Genome Structure: TAD Fusion
 Discovery and Ranking 259
Linh Huynh and Fereydoun Hormozdiari

Assembly of Long Error-Prone Reads Using Repeat Graphs. 261
Mikhail Kolmogorov, Jeffrey Yuan, Yu Lin, and Pavel Pevzner

A Multi-species Functional Embedding Integrating Sequence
 and Network Structure. 263
*Mark D. M. Leiserson, Jason Fan, Anthony Cannistra, Inbar Fried,
 Tim Lim, Thomas Schaffner, Mark Crovella, and Benjamin Hescott*

Deciphering Signaling Specificity with Deep Neural Networks 266
Yunan Luo, Jianzhu Ma, Yang Liu, Qing Ye, Trey Ideker, and Jian Peng

Integrative Inference of Subclonal Tumour Evolution from Single-Cell and
 Bulk Sequencing Data 269
*Salem Malikic, Katharina Jahn, Jack Kuipers, S. Cenk Sahinalp,
 and Niko Beerenwinkel*

Mantis: A Fast, Small, and Exact Large-Scale Sequence-Search Index 271
*Prashant Pandey, Fatemeh Almodaresi, Michael A. Bender,
 Michael Ferdman, Rob Johnson, and Rob Patro*

Tensor Composition Analysis Detects Cell-Type Specific Associations
 in Epigenetic Studies 274
*Elior Rahmani, Regev Schweiger, Saharon Rosset, Sriram Sankararaman,
 and Eran Halperin*

Assembly-Free and Alignment-Free Sample Identification
 Using Genome Skims 276
*Shahab Sarmashghi, Kristine Bohmann, M. Thomas P. Gilbert,
 Vineet Bafna, and Siavash Mirarab*

Efficient Algorithms to Discover Alterations with Complementary
 Functional Association in Cancer 278
Rebecca Sarto Basso, Dorit S. Hochbaum, and Fabio Vandin

Latent Variable Model for Aligning Barcoded Short-Reads Improves
 Downstream Analyses 280
Ariya Shajii, Ibrahim Numanagić, and Bonnie Berger

ModulOmics: Integrating Multi-Omics Data to Identify Cancer
 Driver Modules. 283
*Dana Silverbush, Simona Cristea, Gali Yanovich, Tamar Geiger,
 Niko Beerenwinkel, and Roded Sharan*

SCIΦ: Single-Cell Mutation Identification via Phylogenetic Inference 285
Jochen Singer, Jack Kuipers, Katharina Jahn, and Niko Beerenwinkel

AptaBlocks: Accelerating the Design of RNA-Based Drug
Delivery Systems 287
Yijie Wang, Jan Hoinka, Piotr Swiderski, and Teresa M. Przytycka

A Unifying Framework for Summary Statistic Imputation 289
Yue Wu, Eleazar Eskin, and Sriram Sankararaman

Characterizing Protein-DNA Binding Event Subtypes in ChIP-Exo Data 291
*Naomi Yamada, William K. M. Lai, Nina Farrell, B. Franklin Pugh,
and Shaun Mahony*

Continuous-Trait Probabilistic Model for Comparing Multi-species
Functional Genomic Data. 293
*Yang Yang, Quanquan Gu, Takayo Sasaki, Julianna Crivello,
Rachel O'Neill, David M. Gilbert, and Jian Ma*

Deep Learning Reveals Many More Inter-protein Residue-Residue Contacts
than Direct Coupling Analysis 295
Tian-Ming Zhou, Sheng Wang, and Jinbo Xu

Author Index 297



Long Reads Enable Accurate Estimates of Complexity of Metagenomes

Anton Bankevich¹(✉) and Pavel Pevzner²

¹ Center for Algorithmic Biotechnology, Institute for Translational Biomedicine,
St. Petersburg State University, Saint Petersburg, Russia
anton.bankevich@gmail.com

² Department of Computer Science and Engineering,
University of California at San Diego, La Jolla, CA, USA

Abstract. Although reduced microbiome diversity has been linked to various diseases, estimating the diversity of bacterial communities (the number and the total length of distinct genomes within a metagenome) remains an open problem in microbial ecology. We describe the first analysis of microbial diversity using long reads without any assumption on the frequencies of genomes within a metagenome (parametric methods) and without requiring a large database that covers the total diversity (non-parametric methods). The long read technologies provide new insights into the diversity of metagenomes by interrogating rare species that remained below the radar of previous approaches based on short reads. We present a novel approach for estimating the diversity of metagenomes based on joint analysis of short and long reads and benchmark it on various datasets. We estimate that genomes comprising a human gut metagenome have total length varying from 1.3 to 3.5 billion nucleotides, with genomes responsible for 50% of total abundance having total length varying from only 40 to 60 million nucleotides. In contrast, genomes comprising an aquifer sediment metagenome have more than two-orders of magnitude larger total length (≈ 840 billion nucleotides).

Keywords: Microbial diversity · Metagenomics · Rare species

1 Introduction

Locey and Lennon [29] recently estimated that Earth is home to as many as 1 trillion microbial species. In contrast, Schloss [42] demonstrated that, despite rapidly increasing sequencing efforts, the retrieval of 16S rRNA genes is approaching saturation. They argued that one-third of the bacterial diversity has already been discovered, implying that Earth is home to only millions of, rather than a trillion, bacterial species. This discrepancy and the emerged controversy [1, 25, 26, 35, 44, 55] illustrate that the challenge of evaluating the bacterial diversity remains unsolved both at the global scale and at a single sample (local) level [51]. However, estimating the number of microbial species

in a *given* sample is a more tractable yet difficult problem in microbial ecology [19, 28, 38, 43]. Regrettably, many such estimates are inaccurate since most species in any metagenome belong to the rare biosphere [20, 31, 48]. Indeed, 16S rRNA libraries often capture only a small fraction of the sample diversity, resulting in large variations of the diversity estimates even for samples from similar environments. For example, the estimates of the number of microbial species in soil samples vary from hundreds [22], to tens of thousands [6, 8] to a million [10].

Since the terms “diversity” and “complexity” have multiple interpretations in microbial ecology, we use the terms *metagenome richness* (the total number of species in a metagenome) and *metagenome capacity* (the total genome length of all species in a metagenome). Metagenome capacity can also be defined as metagenome richness multiplied by the estimated average length of genomes in the sample.

Since estimating the richness and capacity of metagenomes is a fundamental problem in microbial ecology, new approaches for solving this problem are needed. This challenge is further amplified by recent discoveries that linked the reduced diversity to various diseases. For example, reduced bacterial diversity results in an increased frequency of death in the allogeneic stem cell transplantation [50] and represents a biomarker for psoriatic arthritis [41]. On the other hand, increased bacterial diversity is associated with human papillomavirus infections [13] and White Plague Disease in corals [49].

The previous studies of bacterial diversity were primarily based on either *parametric* or *non-parametric* approaches [5, 12, 15, 18, 22, 37]. Various parametric distributions were chosen to approximate the frequency distribution of captured species, and to project them to estimate how many more species must be present in the metagenome. This approach has been challenged since it is unclear which parametric distributions adequately model a given sample [17, 54]. Furthermore, 16S rRNA data represents frequencies of specific PCR products that may differ in abundance relative to the corresponding bacterial species in a metagenome. In addition, they suffer from biases arising from various levels of primer matching in different species, inability to amplify taxa whose 16S rRNA differs from known ones, the variable number of 16S rRNA operons, and presence of highly diverged genomes with nearly identical 16S rRNAs [11, 21, 39].

As Hong et al. [17] discussed, applications of various probability distributions for evaluating the complexity of the metagenome have often been statistically incorrect. Moreover, even if some metagenomes follow a certain (e.g., exponential) distribution of frequencies, others may significantly deviate from this arbitrarily chosen model. For example, there is no reason to believe that the frequencies of species for a soil metagenome and a human microbiome follow the same type of parametric probability distribution.

The alternative non-parametric estimators of microbial diversity [22] require a large database that covers all species in a sample, the condition that is typically violated. As a result, such estimates greatly underestimate the richness of metagenomes. As Lladser et al. [28] noted, most microbial communities have not been sufficiently deeply sampled yet to test the suitability of both parametric and non-parametric models.

With proliferation of metagenomics datasets covering entire genomes, it is important to have an independent way of estimating the richness of metagenomes that is not limited to sampling of 16S rRNA. Also, all previous studies attempted to estimate the metagenome richness/capacity using *short* reads that have limitations with respect to analyzing rare species. Since rare species within a metagenome typically have low coverage, they are hardly ever assembled into long contigs, making it difficult to estimate the richness, capacity, and the distribution of frequencies of various species within a metagenome. For example, bacterial species with coverage below 15X typically result in low-quality assemblies, making it difficult to estimate the number of such species or their total genome length. Since such rare species account for the lion's share of genomes in many metagenomes [20], they remain below the radar of modern sequencing technologies.

Below we describe the first analysis of microbial diversity using long reads without any assumption about the distribution of frequencies of genomes within a metagenome (parametric methods) and without requiring a database that covers the total diversity (non-parametric methods).

Our approach to estimating the capacity of a metagenome uses joint analysis of short reads (such as Illumina reads) with long reads (such as TruSeq Synthetic Long Reads, Pacific Biosciences reads, or Oxford Nanopores reads) and rests on a new insight based on *geometric probability* arguments rather than on merely applying the previously proposed approaches to SLRs. We view each long read as a "subgenome" and map all short reads to each subgenome to estimate its abundances. Afterwards, we apply geometric probability arguments to estimate the capacity of the entire metagenome from the abundances of all its subgenomes. We emphasize that our new approach requires *both* long and short reads (i.e., it does not work for short reads only or for long reads only) and demonstrate that it estimates the metagenome capacity and accounts for rare species even if their coverage by reads is below 0.01X, i.e., the species that are not "seen" by the state-of-the-art metagenome assemblers aimed at short reads. We apply our approach to estimate the complexity of the human gut metagenome (in a healthy individual and in multiple samples from a Crohn's disease patient at various stages of the disease) and in an aquifer sediment metagenome.

Although long reads are still rarely used for analyzing metagenomes, they have a potential to be widely used in future metagenomics projects when their cost reduces or when the *read until* technology [30] developed by Oxford Nanopores becomes widely available. We illustrate our approach using the TruSeq *Synthetic Long Reads (SLR)* technology that represents the first long read technology successfully applied to metagenomic studies [46]. The SLR technology generates accurate long virtual reads [27, 32, 52] that provided new insights into diversity of low abundance species in various metagenomes and revealed complex sub-partitioning of metagenomes into dozens of strains of the same species [23, 46, 53]. In addition, SLRs extend 16S rRNA studies, aimed at analyzing taxonomic diversity, by insights into functional diversity of rare species that often provide ecological impact through highly expressed transcriptomes and proteomes [20].

Although this paper focuses on SLRs, our method for estimating the capacity of a metagenome is applicable to long error-prone Single Molecule Sequencing (SMS) reads as well (see Appendix “Estimating metagenome capacity using long error prone SMS reads”).

2 Results

Defining the Capacity of a Metagenome. Figure 1 (left) shows the histogram of frequencies of genomes comprising the artificial MOCK metagenome (with richness 20 and capacity 67 Mb) formed by mixing DNA from 20 isolate genomes [23]. However, the standard measures of richness and capacity depend on the taxonomic definition of a species and do not account for fragments shared by different species within a metagenome and for the fact that some species are represented by multiple similar strains. For example, if a metagenome contains two strains that share 99% of their genomes, should we count them as two genomes (and sum up their lengths) or as a single genome? Similarly, if a genome has a plasmid with multiplicity 100, should we count this plasmid 100 times towards the genome length or just one time?

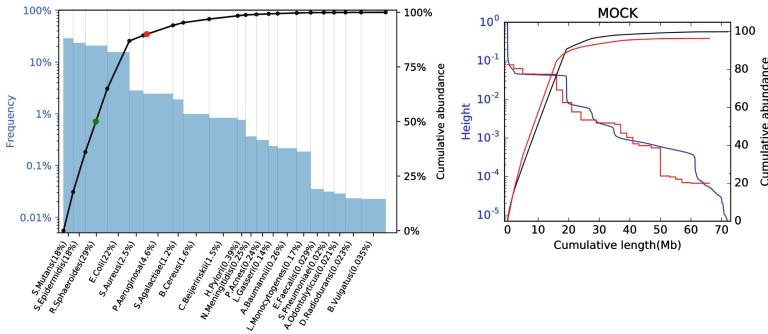


Fig. 1. Real (left) and estimated (right) real frequencies of genomes and the abundance plot (left) and estimated frequency (height) and abundance plot (right) for the MOCK dataset formed by 20 bacterial species. (Left) frequencies (in the logarithmic scale) of genomes in the MOCK dataset vary from $\approx 0.02\%$ to $\approx 28\%$. The green and red points on the abundance plot correspond to $M50 \approx 7$ Mb and $M90 \approx 17.5$ Mb, respectively. The numbers next to the species names represent abundances (note that ranking based on frequencies differs from ranking based on abundances). The genome abundance is approximated as the fraction of short reads originating from this genome. The genome frequency is computed as the genome abundance normalized by the genome length. (Right) Blue and black curves show the estimated frequency (height) histogram and the abundance plot as predicted by our methods in comparison with red curves constructed using the known set of references. (Color figure online)

To address these issues, we introduce the concept of the *de Bruijn capacity* of a metagenome; this is motivated by a popular approach to genome assembly. We

construct the *de Bruijn graph* [7] of all genomes in the metagenome (including plasmids, viruses, unicellular eukaryotes, etc.), transform it into the *assembly graph* using SPAdes [2] (collapsing small variations in repeats), and define the de Bruijn capacity of a metagenome as the total length of edges (contigs) in the assembly graph. The de Bruijn capacity of the MOCK metagenome (58 Mb) is smaller than its total length (67 Mb) since this dataset contains a number of similar genomes and long intragenomic repeats.

Construction of the assembly graph is defined by a parameter *bubble length* that controls the percent identity level used for collapsing regions from various strains into a single contig in the assembly graph. For example, the default bubble length value in the SPAdes assembler [2] roughly corresponds to 98%–99% percent identity with respect to the taxomic definition of a strain (increasing the bubble length parameter will decrease the de Bruijn capacity). Thus, In the default setting, multiple strains that share 98%–99% of the genome (or multicopy plasmids) do not inflate the de Bruijn capacity. To reflect the stringency of taxonomic units in our estimator, one can vary the maximum number of mismatches and indels during alignment of short reads to SLRs.

We characterize each genome G in a metagenome M as a pair of numbers $(length(G), num(G))$, where $length(G)$ and $num(G)$ refer to the length and the copy number of this genome in the metagenome. We define *frequency* and *abundance* of a genome G as

$$frequency(G) = \frac{num(G)}{\sum_{G \in M} num(G)} \quad abundance(G) = \frac{length(G) \cdot num(G)}{\sum_{G \in M} length(G) \cdot num(G)}$$

We note that the number of reads originating from a given genome within a metagenome is roughly proportional to its abundance rather than frequency (under the assumption of the uniform coverage).

The *frequency histogram* of a metagenome consisting of t genomes is defined by t bars with heights specified by the frequencies of the genomes and varying widths specified by the lengths of the genomes (Fig. 1 left). We define the *abundance plot* of a metagenome (Fig. 1 left) by considering t most frequent genomes within a metagenome and specifying a point $(length_t, abundance_t)$, where $length_t$ stands for the total length of these genomes and $abundance_t$ stands for the total abundance of these genomes (for each value of t). For each percentage x from 0% to 100%, we define the value $t(x)$ as the minimum t such that $abundance_t$ exceeds $x\%$. In analogy to the Nx statistics for genome assembly [14], we define the *Mx statistics* for a metagenome as $length_{t(x)}$. For example, $M50 \approx 7$ Mb and $M90 \approx 17.5$ Mb for the MOCK dataset described below (Fig. 1, left).

Computing the abundance plots for complex microbial communities remains an open problem. Below we show how to construct such plots using the synergy between short and long reads.

The Total Rectangle Length Problem. We will first state an abstract probabilistic problem and later explain how it relates to the problem of estimating the

capacity of a metagenome. Consider a set M of rectangles, each rectangle R in this set characterized by its length $length(R)$ and height $height(R)$. We assume that the total area of rectangles is 1, i.e., $\sum_{R \in M} length(R) \cdot height(R) = 1$.

For a point ξ from one of the rectangles, we define $height(\xi)$ as the height of the rectangle that the point ξ falls into. We uniformly and independently sample N points from the total area of all rectangles and denote the rectangle the j -th point falls into as R_j , which is characterized by its length and height $(length_j, height_j)$ (Fig. 2). We further assume that the vector $(height_1, \dots, height_N)$ is known but the vector $(length_1, \dots, length_N)$ is unknown.

Let ξ be a random variable corresponding to the uniform sampling of a point from the total area of all rectangles. The described probabilistic process results in N samples of the random variable $height(\xi)$. Given a vector $(height_1, \dots, height_N)$, our (somewhat ambitious) goal is to estimate the total length of all rectangles: $Length(M) = \sum_{R \in M} length(R)$.

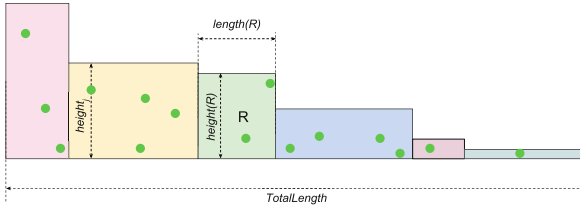


Fig. 2. The Total Rectangle Length Problem. N green points are sampled uniformly and independently from the probabilistic space defined by an (unknown) set of rectangles. Assuming that the heights of rectangles $(height_1, \dots, height_N)$ these points fall into are known, the goal is to estimate the total length of all rectangles. (Color figure online)

The Total Rectangle Length Problem is intractable since the set of rectangles may contain a myriad of rectangles with extremely small heights and huge lengths whose total area is very small, e.g., significantly below $1/N$. Since these rectangles are unlikely to be sampled by any of N sampled points, our estimate cannot take into account their total length. We thus assume that all rectangles in the dataset have sufficiently large areas (e.g., larger than $1/N$) so that the probability of sampling each rectangle by at least one point is high.

Estimating the Total Length of Rectangles. Since the points are sampled uniformly, the probability of a point falling into a rectangle R equals the area of R : $area(R) = \Pr(\xi \in R)$. Thus, the length of the rectangle R is $length(R) = area(R)/height(R) = \Pr(\xi \in R)/height(R)$ and the total length of all rectangles can be estimated as:

$$Length(M) = \sum_{R \in M} \frac{\Pr(\xi \in R)}{height(R)} = \sum_{R \in M} \int_{x \in R} \frac{dx}{height(x)} = E \left(\frac{1}{height(\xi)} \right), \quad (1)$$

where E stands for the expectation of a random variable. Thus, by the law of large numbers:

$$Length(M) \approx \frac{1}{N} \sum_{j=1}^N \frac{1}{height_j}. \quad (2)$$

Moreover, according to the central limit theorem, the formula above is accurate, i.e., for large N , the variance of the estimate above is approximated as the variance of the random variable $\frac{1}{height(\xi)}$ divided by N . We will use this feature to estimate the metagenome capacity without any assumption on the parametric distribution of frequencies of genomes within a metagenome. See the Methods section for computing the abundance plot using a similar approach.

Representing a Metagenome as a Set of Rectangles. As before, we characterize each genome G in a metagenome M as a pair of numbers $(length(G), num(G))$ and define the total length of all genomes over all cells in a metagenomic sample as

$$sum = \sum_{G \in M} length(G) \cdot num(G)$$

The height of a genome G is defined as $height(G) = num(G)/sum$. Note that genome height is proportional to genome frequency. Thus each genome G is characterized by a rectangle $(length(G), height(G))$ and the total area of all rectangles is 1.

We assume that each genome within a metagenome results in a number of reads that is roughly proportional to its abundance. We also assume that each read is characterized by its starting position in one of the genomes and that these starting positions sample the genome uniformly and independently. Although the depth of coverage may deviate from the mean coverage in some genomic regions (e.g., in GC-rich regions or in the regions close to the origin of replication in the case of actively replicating genomes), such deviations are usually small. For the sake of simplicity, we assume that all genomes are circular, e.g., a read can start at any position of the genome so that there are no borderline artifacts (in the case of linear genomes, there are no reads that start within the last $i - 1$ positions of the genome, where i is the read length). Thus, starting positions model the random variable corresponding to the uniform sampling of a point from the total area of all rectangles.

We assume that N_{long} long reads and N_{short} short reads were sampled from the metagenome and that N_{short} is much larger than N_{long} . For example, various samples we analyzed contained ≈ 30 – 100 million short paired-end reads and ≈ 100 – 800 thousand long SLRs.

Each long defines a random point within the set of rectangles and our goal is to estimate the height of the rectangle this point falls into. We use short Illumina reads mapping to a given long read to estimate this height.

For simplicity, we assume that all short reads have the same length referred to as $|shortRead|$ (this condition holds for most sequencing projects). Since SLRs

are accurate, we assume that each short read aligned to an SLR *longRead* also maps to the corresponding genomic segment and vice versa. To avoid borderline effects, we assume that we can detect all short reads that align to *longRead*, even reads that start at the last position of *longRead*. To satisfy this condition, we shorten each SLR by $|shortRead|$ (or by the insert length) but map short reads using the entire span of each SLR. We define the number of short reads mapping to an SLR *longRead* as $number(longRead)$.

The fraction of short reads that map to *longRead* is expected to be approximately equal to the area in the rectangle space “above” *longRead*, i.e., to $|longRead| \cdot height(longRead)$, where $height(longRead)$ is defined as the height of the rectangle (genome) that contains *longRead*. Thus, the expected number of short reads that map to *longRead* (that we refer to as $E(number(longRead))$) can be estimated as

$$E(number(longRead)) = |longRead| \cdot height(longRead) \cdot N_{short} \quad (3)$$

Thus,

$$height(longRead) \approx \frac{number(longRead)}{N_{short} \cdot |longRead|}. \quad (4)$$

We just reduced the problem of estimating the capacity of a metagenome to the Total Rectangle Length Problem. We are given a set of N_{long} points (SLRs) that represent a uniform and independent sampling of an unknown set of rectangles (the metagenome). Each SLR is characterized by its length $|longRead_j|$ and the number of short reads $number_j$ mapping to this SLR (for $1 \leq j \leq N_{long}$). We estimate the height h_j of the j -th SLR read using formula 4. Thus, the capacity of the metagenome is estimated as

$$Capacity(Metagenome) \approx \frac{1}{N_{long}} \sum_{j=1}^{N_{long}} \frac{1}{height_j} \approx \frac{N_{short}}{N_{long}} \sum_{j=1}^{N_{long}} \frac{|longRead_j|}{number_j}. \quad (5)$$

In the Methods section we describe similar formulas for constructing the frequency histogram and the abundance plots. Formula 5 has limitations in analyzing extremely rare species, e.g., species that did not result in any SLRs or species that resulted in SLRs with extremely small coverage by short reads. Note that this formula was derived in two steps: estimation of the expectation of the inverse height (formula 2) and estimation of the SLR height through its coverage by short reads (formula 4). We discuss how to address the limitations of these steps in the Methods section.

Datasets. We analyzed the following metagenomics datasets based on SPAdes [2] and truSPAdes [3] assemblies of SLRs (see Appendix “TruSPAdes assemblies of MOCK, GUT, and SEDI datasets”):

The SYNTH synthetic community dataset is formed by a set of short Illumina reads from the genomic DNA mixture of 64 diverse bacterial and archaeal species (Shakya et al. [45]; SRA acc. no. SRX200676) that was used for benchmarking

the Omega assembler [16]. It contains ≈ 109 million Illumina HiSeq 100 bp paired-end reads with mean insert size of 206 bp. Since the reference genomes for all 64 species forming the SYNTH dataset are known, we used them to evaluate the accuracy of our estimator. The total length of the reference genomes for this dataset is ≈ 200 Mb and its de Bruijn complexity is ≈ 190 Mb.

The SYNTH dataset contains short Illumina reads but does not contain SLRs. We thus simulated 6306 virtual SLRs (providing the average coverage of 0.25 for the metagenome) for the SYNTH dataset by randomly selecting a short read, mapping it to one of the reference genomes, and extending the region covered by this read by N nucleotides in both directions (N was uniformly distributed between 1500 to 5500). This simulation protocol ensures that simulated SLRs are sampled from metagenome with the same probability distribution as the short reads.

The MOCK synthetic community dataset is formed by short Illumina reads and SLRs from the genomic DNA mixture of 20 bacterial species [23]. It contains ≈ 31 million Illumina paired-end short reads with mean insert size of 247 bp and ≈ 221 thousand SLR reads longer than 6 kb constructed from three sets of 384 barcoded read pools each. Since the reference genomes for all species forming the MOCK dataset are known, we used them to assess the accuracy of our estimator. The total length of the reference genomes for this dataset is ≈ 67 Mb and its de Bruijn complexity is ≈ 58 Mb.

The GUT dataset is formed from short Illumina reads and SLRs sampled from the gut microbiome of a healthy human male that was analyzed in Kuleshov et al. [23]. It contains ≈ 80 million paired-end short reads with mean insert size of 208 bp and seven sets of barcoded read pools (384 pools in each set) that resulted in ≈ 501 thousand SLR contigs longer than 6 kb. Using this dataset we provide a new estimate of the capacity of the human gut metagenome.

The SEDI dataset is formed from short Illumina reads and SLRs sampled from an aquifer sediment that was analyzed in Sharon et al. [46]. It contains ≈ 27 million paired-end short reads with mean insert size of 351 bp and three sets of barcoded read pools (384 pools in each set) that resulted in ≈ 215 thousand SLRs longer than 6 kb. Sharon et al. [46] revealed a high diversity of strains in the genomes of this dataset. We confirm findings of Sharon et al. [46] and turn their initial observation into an estimate of the SEDI metagenome capacity.

In difference from the SYNTH and MOCK datasets, the metagenome capacity of GUT and SEDI datasets remains unknown. In addition to these datasets, we analyzed a larger synthetic dataset and four human microbiome datasets from a patient suffering from the Crohn’s disease (see Appendix).

Benchmarking. For each dataset, we estimated the capacity of the corresponding metagenome (using formula 2) and constructed the frequency histogram and the abundance plot (Fig. 1 (right) for MOCK dataset and Fig. 3 for the other three datasets) using formula 7. We analyzed 1000 SLRs with the highest coverage in each dataset (contributing to a small “bump” in the beginning of the frequency histograms) and confirmed that most of them arise from plasmids and

16S rRNAs. This finding suggests that highly-covered SLRs can be used for *de novo* assembly of new plasmids and characterization of previously unknown 16S rRNAs directly from metagenomics datasets. Recent attempts to address these problems using short reads with tools like RECYCLER [40] and PhylOTU [47] faced computational challenges since it remains unclear how to extract plasmids and 16 rRNAs from the complex de Bruijn graphs of metagenomes. In order to evaluate how our estimator deteriorates with reduction in coverage by short reads and/or long reads, we downsampled the entire datasets of short reads and SLRs (Table 1).

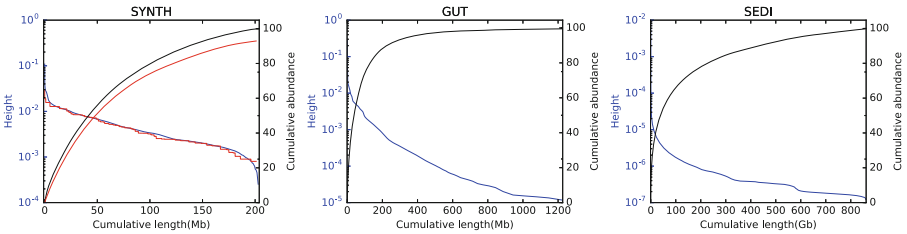


Fig. 3. Estimated frequency histograms (blue curve) and abundance plots (black curve) for SYNTH, GUT, and SEDI datasets. The distribution of heights (frequencies) of individual genomes within a metagenome was obtained based on alignments of short reads to SLRs. For the SYNTH dataset, we compared the constructed frequency histogram and abundance plot with the red plot representing the reference genomes with known abundancies. The y -axis of frequency histograms show the histogram of heights SLRs (in the decreasing order of heights) multiplied by 10^6 , i.e., the probability that a random read falls into a 1 Mb long segment of the metagenome specified by the x coordinate. For the GUT dataset, $M50 = 40$ Mb and $M90 = 230$ Mb. For the SEDI dataset, $M50 = 39$ Gb and $M90 = 432$ Gb. (Color figure online)

SYNTH. As Table 1 illustrates, our estimator is accurate even with a small number of SLRs and short reads; e.g., even for short reads downsampled at 5%, deviation from the total metagenome size does not exceed 15%.

Note that the coverage of some genomes in the SYNTH dataset is as low as 6X [16]. Our capacity estimate remains accurate even at 0.1% downsampling, corresponding to the coverage by short reads for some genomes as low as 0.006. Note that the estimated capacity is accurate when almost all SLRs are covered by at least one short read.

MOCK. Table 1 illustrates that our formula accurately estimates the metagenome capacity when at least 7% of short reads are used. Note that while the MOCK dataset is subject to various biases that affect sampling of SLR and short reads (e.g., the GC bias), our formula is still accurate. Table 1 also shows that our formula generates stable capacity estimates even with a highly variable number of downsampled SLRs and suggests that there is a number of rare species in this metagenome.

Table 1. The metagenome complexity estimation (in Mb) for **SYNTH** and **MOCK** datasets. Columns correspond to downsampling of SLR reads, while rows correspond to downsampling of short reads. The last column shows the percentage of SLRs that were not covered by any reads from the downsampled set of short reads. Estimated metagenome capacity (in Mb) for **SYNTH** and **MOCK** datasets is 200 Mb and 67 Mb, respectively. Estimated de Bruijn capacity (in Mb) for **SYNTH** and **MOCK** datasets is 190 Mb and 58 Mb, respectively.

SYNTH					
	Estimated metagenome capacity using				Fraction of uncovered SLRs
	100 SLRs	500 SLRs	2000 SLRs	10000 SLRs	
0.02%	156	144	147	150	51%
0.1%	204	220	209	218	28%
1%	194	241	224	230	0.4%
5%	182	221	203	205	0%
25%	179	212	198	201	0%
100%	179	209	196	199	0%
MOCK					
	Estimated metagenome capacity using				Fraction of uncovered SLRs
	100 SLRs	1000 SLRs	10000 SLRs	220748 SLRs	
1%	34	31	37	37	3%
7%	53	56	59	58	0.7%
20%	76	73	73	71	0.08%
100%	69	60	68	72	0.005%

GUT. We estimated the capacity of the human gut metagenome at ≈ 1.3 billion nucleotides, in line with previous estimates of the human gut microbiome richness [36]. Also, a rather small fraction of SLRs were not covered by reads (0.8%), suggesting that our estimate is accurate. Note that assembly of this dataset performed in Kuleshov et al. [23] resulted in contigs with total length of 656 Mb. Thus, the assembled contigs in the **GUT** dataset represent a large fraction of this metagenome.

SEDI. Our formula resulted in ≈ 840 Gb estimate for the capacity of this metagenome but $\approx 47\%$ of SLRs were not covered by any short reads, suggesting that this metagenome is very diverse and that it contains a very large number of extremely rare species (with coverage 0.01X and below) which account for most of the total DNA in this metagenome. Thus, our formula is likely to underestimate the complexity of this metagenome. Note that the total length of assembled contigs for the **SEDI** dataset (204 Mb for contigs longer than 1 kb) is significantly lower than the estimated capacity of the metagenome. Since the large **SEDI** metagenome may include unicellular eukaryotes with large genomes (that are common in sediments [4]) and is likely to include a large fraction of relic DNA [26], it is difficult to estimate its richness.

3 Methods

Estimating the Abundance Plot. Let D be a value range of a random variable ξ with density p with respect to a measure μ . By considering $p(\xi)$ as a random variable, we have:

$$E\left(\frac{1}{p(\xi)}\right) = \int_D \frac{1}{p(x)} p(x) = \int_D 1 = |D|. \quad (6)$$

Thus, formula 1 is a special case of a more general formula for the value range size estimation. This interpretation also allows us to estimate the value of $|D_t|$, where $D_t = \{x \in D | p(x) < t\}$:

$$|D_t| = \int_{D_t} 1 = \int_{D_t} \frac{1}{p(x)} p(x) = \int_D \frac{1}{p(x)} \delta_{p(x) < t} p(x) = E\left(\frac{1}{p(\xi)} \cdot \delta_{p(\xi) < t}\right). \quad (7)$$

The right part of this formula can be estimated similarly to formula 2, resulting in the estimate of the frequency histogram. The graph of $|D_t|$ as a function of t gives the abundance plot of a metagenome. In practice estimation of frequency histogram can be constructed using the following method. Given the heights of SLRs in the decreasing order $(h_1, \dots, h_{N_{long}})$ computed using formula 4, the frequency histogram consists of N_{long} bars with the j -th bar in the histogram having height h_j and width $\frac{1}{h_j}$. The abundance plot is merely the integral of the frequency histogram.

Variance of the Metagenome Capacity Estimator. We used the central limit theorem (CLT) as the basis of our estimator. The accuracy of the resulting formula in the CLT is defined by the variance of the random variable in question. For example, in the case when a significant fraction of a metagenome results in rectangles with extremely low height (e.g., rectangles with area less than $1/N_{long}$), the variance of the random variable is very high. We thus make an assumption that nearly entire metagenome is comprised from the genomes with sufficiently large frequencies to be captured by SLRs. Since typical SLR projects result in 10^5 – 10^6 SLRs, this constraint implies that the metagenome that we are able to analyze mostly consists of species with frequencies exceeding 0.001%. Under this assumption, we can use the CLT to compute the variance of our estimator.

Accuracy of the Inverse Height Estimator. Formula 3 leads to an unbiased estimate of the SLR height $height(longRead)$ (given by formula 4). However, the value that we actually need to estimate is $\frac{1}{height(longRead)}$ and this estimation, given by formula 5, becomes biased. Below we analyze how this bias affects our estimation of the metagenome capacity.

We first consider a simple case when the metagenome consists of a single genome *Genome* and when SLRs sampled from *Genome* have the same length.

We also assume that the number of reads mapped to a genome fragment (and an SLR) follows the Poisson distribution:

$$\text{number}(\text{longRead}) \sim \text{Poisson}(\lambda), \quad (8)$$

where λ represents the expectation of the number of reads mapped to *longRead*. The value λ can be estimated as: $\lambda = |\text{longRead}| \cdot \text{height}(\text{longRead}) \cdot N_{\text{short}}$. We can now compute the value $|\text{Genome}|^*$, the genome length that is (erroneously) estimated by formula 5 instead of $|\text{Genome}|$:

$$|\text{Genome}|^* = N_{\text{short}} \cdot E \left(\frac{|\text{longRead}|}{\text{Poisson}(\lambda)} \mid \text{Poisson}(\lambda) \neq 0 \right) \quad (9)$$

Note that since $\text{height}(\text{Genome}) \cdot |\text{Genome}| = 1$, the function δ , defined as $\frac{|\text{Genome}|^*}{|\text{Genome}|}$, depends only on the value of λ :

$$\begin{aligned} \delta(\lambda) &= \frac{|\text{Genome}|^*}{|\text{Genome}|} = \lambda \cdot E \left(\frac{1}{\text{Poisson}(\lambda)} \mid \text{Poisson}(\lambda) \neq 0 \right) \\ &= \frac{\lambda}{1 - e^{-\lambda}} \sum_{n=1}^{\infty} [(1/n) \cdot e^{-\lambda} \cdot \lambda^n / n!] = \frac{\lambda \cdot e^{-\lambda}}{1 - e^{-\lambda}} (-\gamma - \ln(\lambda) - Ei(-\lambda)) \end{aligned}$$

where $\gamma \approx 0.57721566$ is the *Euler-Mascheroni* constant and *Ei* is the *exponential integral* $Ei(z) = -\int_{-z}^{\infty} e^{-t} t^{-1} dt$. Thus, the expectation of the relative error in formula 5 is defined by $\delta(\lambda)$. The higher is the value of λ (which refers to the average number of short reads mapped to a long read), the closer δ is to 1. For example, if the expected number of short reads aligned to an SLR exceeds 15, the relative error of our estimate is at most 10%. Coverage of a typical 10 kb long SLR by 15 reads corresponds to genome coverage of $15 \cdot |\text{shortRead}| / |\text{longRead}| = 0.15X$ for short reads of length 100.

This analysis illustrates why long reads provide a much “deeper” look into the capacity of a metagenome than short reads. Indeed, it enables analysis of genomes with the coverage 0.15X and below as compared to the coverage 15X that is typically needed for assembling a genome within a metagenome from short reads. For genomes with a value of λ significantly less than 1, it turns out that most SLRs sampled from them have zero coverage by short reads. Thus, genomes with very low coverage contribute little to the estimate of the metagenome capacity.

4 Discussion

The recent bacterial census update [42] highlighted that high-throughput sequencing is based on short reads, while a high-quality census requires a high-throughput *full-length* 16S rRNA sequencing (rather than conventional short reads sequencing). It also illustrated the need for alternative technologies to analyze bacterial diversity such as single cell sequencing [21]. However, without

prior sorting, single cell sequencing mostly reports the abundant species. In contrast, a large fraction of individual genomes assembled from metagenomes had not been sequenced before [34]. However, the number of genomes reliably recovered from a metagenome is usually limited to hundreds at best, a small fraction of the total diversity of a metagenome. These difficulties highlight the need for a yet another technology for evaluating bacterial diversity. We showed that a combination of short-read and long-read sequencing technologies solves this problem even though each of these technologies separately does not provide accurate estimates of the metagenome capacity. Although our analysis may be hampered by a potential metagenome sampling bias between short and long reads, our estimator of a metagenome complexity results in a useful approximation of the metagenome size.

Analysis of various metagenomics samples revealed that, although there often exists a small number of abundant species, thousands of low-abundance highly-diverged species account for most of the observed diversity. While this rare biosphere represents a source of genomic innovation [20], previous metagenomics studies, plagued by limitations of short reads technologies, were unable to evaluate its diversity. This study is the first attempt to estimate the diversity of the rare biosphere using a combination of short and long reads. Our analysis of the SEDI dataset illustrates, this rare biosphere may contain hundreds of thousands species even for a single soil sample. As the existing estimates of richness of soil and sediment bacterial communities differ by orders of magnitudes, it would be interesting to apply our approach to analyzing other soil/sediment hybrid datasets when they become available.

Our approach also revealed significant variations in the diversity of the human gut metagenome in the case of an individual with the Crohn’s disease. We envision that the metagenomics studies will soon move to generating a nearly complete census of all bacteria within microbiomes across the entire human population [33]. Our method will provide an estimate of the still unknown fraction of metagenomes that will be important for building such a census.

Acknowledgements. We are indebted to Chris Dupont, Rob Knight, and Glenn Tesler for providing numerous comments. Glenn Tesler also suggested using exponential integrals for analyzing the bias of our estimator. We are grateful to Yana Safonova, Andrey Bzikadse, Sergey Bankevich, Sergey Nurk, Alon Orlitsky, Ivan Tolstoganov, and Aleksandr Shlemov for many helpful discussions and help with preparation of this paper. This study was funded by the Russian Science Foundation (award 14-50-00069) and by the National Science Foundation (MCB-BSF award 1715911).

Appendix

TruSPAdes Assemblies of MOCK, GUT, and SEDI Datasets. The TruSeq SLR technology generates accurate and long virtual reads derived from pools of short reads [27, 32, 52]. It is based on fragmenting genomic DNA into large segments (≈ 10 kb long) and forming random pools of the resulting segments (each pool contains ≈ 300 segments). Next, these fragments are amplified,

sheared, and marked with a barcode that is unique to the pool. Afterwards, they are sequenced using the standard Illumina short reads technology. All short reads originating from the same barcode are assembled together resulting in a set of long contigs (this step is called the *SLR barcode assembly*). Ideally, the result of such sequencing effort for a single barcode is the collection of approximately 300 fragments (each fragment is ≈ 10 kb long) from a genome forming 300 long virtual reads. SLRs have low mismatch rate (about 0.1%), extremely low indel rate, and few misassemblies [3].

Table 2 presents results of barcode assembly of MOCK, GUT and SEDI datasets with truSPAdes.

Table 2. Results of truSPAdes assemblies of MOCK, GUT and SEDI datasets. Long SLRs are defined as SLRs longer than 6 kb.

	MOCK	GUT	SEDI
#SLRs	451036	1226918	210495
#long SLRs	220778	772833	157336
N50	9180	8625	8266
Avg. #long SLRs per barcode	191	287	136
Total length of SLRs (Gb)	2.9	8.4	1.5
Total length of long SLRs (Gb)	2.1	5.8	1.3

Analyzing the CAMI and CROHN Datasets. In addition to datasets described in the main text, we also analyzed a larger synthetic dataset and four human microbiome datasets from a patient suffering from the Crohn’s disease.

The CAMI dataset is a simulated dataset generated by the “Critical Assessment of Metagenome Interpretation” (CAMI) initiative aimed at evaluating various approaches to analyzing metagenomes (<http://www.cami-challenge.org/>). We used a CAMI dataset simulated from 225 genomes and containing 150 million 100bp paired-end reads with mean insert size of 180bp (the errors in simulated reads are modelled after Illumina HiSeq). We simulated 50 thousand SLRs in the same way as for the SYNTH dataset. The total length of the reference genomes for this dataset is ≈ 820 Mb and its de Bruijn complexity is ≈ 770 Mb. Figure 4 shows that our estimator works well for the CAMI dataset.

The CROHN datasets are four human gut microbiome datasets from a patient with Crohn’s disease. These datasets (CROHN1, CROHN2, CROHN3, CROHN4) represent a metagenomics time series collected at 12-28-2011, 04-29-2013, 11-16-2014 and 06-29-2015, respectively. Each of these datasets includes one Illumina paired-end library and one SLR library. Number of short reads in these datasets ranges from 150 to 230 millions with mean insert size ≈ 400 bp for all datasets. The number of SLRs ranges from 17 to 50 thousand. Assembly efforts for these datasets resulted in contigs of length 242, 172, 225 and 275 Mb for CROHN1, CROHN2, CROHN3, and CROHN4 datasets respectively.

We estimated metagenome capacity for CROHN1, CROHN2, CROHN3, and CROHN4 datasets as 3.5, 2.0, 2.4, and 3.2 Gb, respectively. Values of M50 were estimated as 41, 61, 25, and 45 Mb, respectively, while values of M90 were estimated as 230, 490, 240, 250 Mb respectively. These estimates reveal large variations in metagenome capacity during the course of disease that go well beyond what can be estimated using short read assemblies. Correlation between metagenome capacity and antibiotics treatments for this metagenomics time series will be discussed elsewhere.

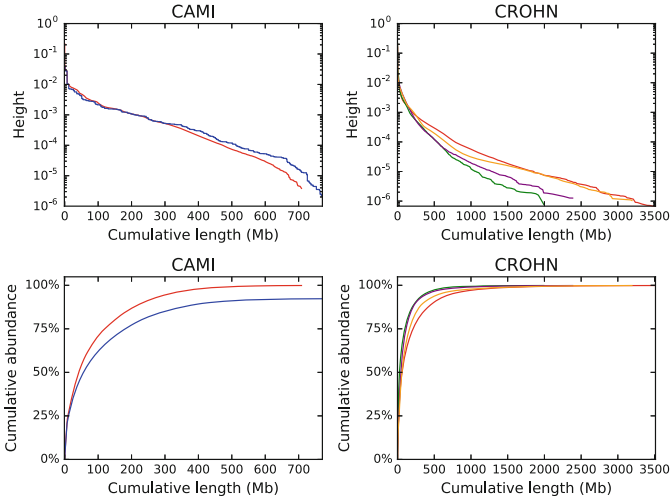


Fig. 4. Estimated frequency histograms and abundance plots for CAMI (left) and CROHN1, CROHN2, CROHN3, CROHN4 datasets (right). The distribution of heights (frequencies) of individual genomes within a metagenome was obtained based on alignments of short reads to SLRs. For the CAMI dataset, we compared the constructed plots with the blue plot representing the reference genomes with known abundancies.

Estimating Metagenome Capacity Using Long Error Prone SMS Reads. Although SMS reads (e.g., reads generated using Pacific Biosciences and Oxford Nanopores technologies) are still rarely used for analyzing metagenomes [9], they have a potential to be widely used in future metagenomics projects when their cost reduces and when the *read until* technology [30] developed by Oxford Nanopores becomes widely available. Below we show how to extend our approach for estimating the metagenome complexity using SMS reads.

SMS reads present an attractive alternative to TSLRs since their average length is higher and since they feature a uniform coverage depth that is not affected by the GC content. However, alignment of short Illumina reads against error-prone SMS reads is a more challenging task than their alignment against accurate TSLRs. We addressed this complication using the bowtie2 alignment tool [24] with specially selected parameters aimed at alignment of short Illumina

reads against error-prone SMS reads (-D 40 -R 3 -N 0 -L 17 -i S,1,0.50 -rdg 1,3 -rfg 1,3 -score-min L,-0.6,-1 -a). However, even using these custom parameters, bowtie2 fails to detect alignments of $\approx 20\%$ of Illumina reads, resulting in an underestimation of the heights of long reads. To compensate for this effect, we applied an adjustment factor $\frac{100}{100-20} = 1.25$ to artificially inflate the heights in our formula for estimating the metagenome capacity.

Currently, there is a shortage of publicly available hybrid metagenomics datasets (containing both Illumina and SMS reads). Ideally, Illumina and SMS reads for such datasets should be generated at the same time so that the abundances of individual genomes within a metagenome are the same for Illumina and SMS reads, implying that the depth of coverage by Illumina reads is proportional to the depth of coverage by SMS reads. In practice, since the SMS reads for these datasets were often generated as an afterthought, Illumina and SMS reads for the publicly available hybrid metagenomics datasets are generated at different time points and prepared for sequencing using different sample preparation protocols. Thus, since metagenome composition is changing and is subject to blooms [33], the existing hybrid datasets do not necessarily feature the proportional depths of coverage by Illumina and SMS reads. Our analysis revealed that the fractions of Illumina and SMS reads aligned to each of the reference genomes for publicly available hybrid synthetic metagenomic dataset may differ by two orders of magnitude. This difference in the genome coverages by short and long reads in the publicly available hybrid metagenomics datasets makes our approach inapplicable to the currently available hybrid metagenomics datasets.

References

1. Amann, R., Rosselló-Móra, R.: After all, only millions? *mBio* **7**(4), e00,99916 (2016)
2. Bankevich, A., Nurk, S., Antipov, D., et al.: SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.* **19**(5), 455–477 (2012)
3. Bankevich, A., Pevzner, P.A.: TruSPAdes: barcode assembly of TruSeq synthetic long reads. *Nat. Methods* **13**, 248–250 (2016). <https://doi.org/10.1038/nmeth.3737>
4. Capo, E., Debroas, D., Arnaud, F., Domaizon, I.: Is planktonic diversity well recorded in sedimentary DNA? Toward the reconstruction of past protistan diversity. *Microb. Ecol.* **70**(4), 865–875 (2015)
5. Chao, A., Bunge, J.: Estimating the number of species in a stochastic abundance model. *Biometrics* **58**(3), 531–539 (2002). <https://doi.org/10.1111/j.0006-341X.2002.00531.x>
6. Chen, Y., Kuang, J., Jia, P., Cadotte, M.W., Huang, L., Li, J., Liao, B., Wang, P., Shu, W.: Effect of environmental variation on estimating the bacterial species richness. *Front. Microbiol.* **8**, 690 (2017)
7. Compeau, P.E.C., Pevzner, P.A., Tesler, G.: How to apply de Bruijn graphs to genome assembly. *Nat. Biotechnol.* **29**(11), 987–991 (2011). <https://doi.org/10.1038/nbt.2023>






8. Curtis, T.P., Sloan, W.T., Scannell, J.W.: Estimating prokaryotic diversity and its limits. *Proc. Natl. Acad. Sci. U.S.A.* **99**(16), 10494–10499 (2002). <https://doi.org/10.1073/pnas.142680199>
9. Driscoll, C.B., Otten, T.G., Brown, N.M., Dreher, T.W.: Towards long-read metagenomics: complete assembly of three novel genomes from bacteria dependent on a diazotrophic cyanobacterium in a freshwater lake co-culture. *Stand. Genomic Sci.* **12**(1), 9 (2017)
10. Dykhuizen, D.E.: Santa Rosalia revisited: why are there so many species of bacteria? *Antonie Van Leeuwenhoek* **73**(1), 25–33 (1998)
11. Ellegaard, K.M., Engel, P.: Beyond 16S rRNA community profiling: intra-species diversity in the gut microbiota. *Front. Microbiol.* **7**, 1475 (2016)
12. Frisli, T., Haverkamp, T.H.A., Jakobsen, K.S., Stenseth, N.C., Rudi, K.: Estimation of metagenome size and structure in an experimental soil microbiota from low coverage next-generation sequence data. *J. Appl. Microbiol.* **114**(1), 141–151 (2013). <https://doi.org/10.1111/jam.12035>
13. Gao, W., Weng, J., Gao, Y., Chen, X.: Comparison of the vaginal microbiota diversity of women with and without human papillomavirus infection: a cross-sectional study. *BMC Infect. Dis.* **13**(1), 271 (2013)
14. Gurevich, A., Saveliev, V., Vyahhi, N., Tesler, G.: QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**(8), 1072–1075 (2013). <https://doi.org/10.1093/bioinformatics/btt086>
15. Haegeman, B., Hamelin, J., Moriarty, J., Neal, P., Dushoff, J., Weitz, J.S.: Robust estimation of microbial diversity in theory and in practice. *ISME J.* **7**(6), 1092–1101 (2013). <https://doi.org/10.1038/ismej.2013.10>
16. Haider, B., Ahn, T.H., Bushnell, B., et al.: Omega: an overlap-graph de novo assembler for metagenomics. *Bioinformatics* **30**(19), 2717–2722 (2014). <https://doi.org/10.1093/bioinformatics/btu395>
17. Hong, S.H., Bunge, J., Jeon, S.O., Epstein, S.S.: Predicting microbial species richness. *Proc. Natl. Acad. Sci. U.S.A.* **103**(1), 117–122 (2006). <https://doi.org/10.1073/pnas.0507245102>
18. Hooper, S.D., Dalevi, D., Pati, A., Mavromatis, K., Ivanova, N.N., Kyrpides, N.C.: Estimating DNA coverage and abundance in metagenomes using a gamma approximation. *Bioinformatics* **26**(3), 295–301 (2010). <https://doi.org/10.1093/bioinformatics/btp687>
19. Hughes, J.B., Hellmann, J.J., Ricketts, T.H., Bohannan, B.J.: Counting the uncountable: statistical approaches to estimating microbial diversity. *Appl. Environ. Microbiol.* **67**(10), 4399–4406 (2001)
20. Jousset, A., Bienhold, C., Chatzinotas, A., et al.: Where less may be more: how the rare biosphere pulls ecosystems strings. *ISME J.* **33**(4), 853–862 (2017)
21. Kashtan, N., Roggensack, S.E., Rodrigue, S., et al.: Single-cell genomics reveals hundreds of coexisting subpopulations in wild prochlorococcus. *Science* **344**(6182), 416–420 (2014)
22. Kemp, P.F., Aller, J.Y.: Bacterial diversity in aquatic and other environments: what 16S rDNA libraries can tell us. *FEMS Microbiol. Ecol.* **47**(2), 161–177 (2004). [https://doi.org/10.1016/S0168-6496\(03\)00257-5](https://doi.org/10.1016/S0168-6496(03)00257-5)
23. Kuleshov, V., Jiang, C., Zhou, W., Jahanbani, F., Batzoglou, S., Snyder, M.: Synthetic long-read sequencing reveals intraspecies diversity in the human microbiome. *Nat. Biotechnol.* **34**(1), 64–69 (2015). <https://doi.org/10.1038/nbt.3416>
24. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**(4), 357–359 (2012)

25. Lennon, J.T., Locey, K.J.: The underestimation of global microbial diversity. *mBio* **7**(5), e01,298-16 (2016). <https://doi.org/10.1128/mBio.01298-16>
26. Lennon, J.T., Placella, S.A., Muscarella, M.E.: Relic DNA contributes minimally to estimates of microbial diversity. *bioRxiv*, p. 131284 (2017)
27. Li, R., Hsieh, C.L., Young, A., et al.: Illumina synthetic long read sequencing allows recovery of missing sequences even in the “finished” *C. elegans* genome. *Sci. Rep.* **5**, 10,814 (2015). <https://doi.org/10.1038/srep10814>
28. Lladser, M.E., Gouet, R., Reeder, J.: Extrapolation of urn models via poissonization: accurate measurements of the microbial unknown. *PLoS ONE* **6**(6), e21,105 (2011). <https://doi.org/10.1371/journal.pone.0021105>
29. Locey, K.J., Lennon, J.T.: Scaling laws predict global microbial diversity. *Natl. Acad. Sci. U.S.A.* **113**(21), 5970–5975 (2016)
30. Loose, M., Malla, S., Stout, M.: Real-time selective sequencing using nanopore technology. *Nat. Methods* **13**(9), 751–754 (2016)
31. Lynch, M.D.J., Neufeld, J.D.: Ecology and exploration of the rare biosphere. *Nat. Rev. Microbiol.* **13**(4), 217–229 (2015). <https://doi.org/10.1038/nrmicro3400>
32. McCoy, R.C., Taylor, R.W., Blauwkamp, T.A., et al.: Illumina TruSeq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PLoS ONE* **9**(9), e106,689 (2014). <https://doi.org/10.1371/journal.pone.0106689>
33. McDonald, D., et al.: American gut: an open platform for citizen-science microbiome research (2018, submitted)
34. Miller, C.S., Baker, B.J., Thomas, B.C., Singer, S.W., Banfield, J.F.: Emirge: reconstruction of full-length ribosomal genes from microbial community short read sequencing data. *Genome Biol.* **12**(5), R44 (2011). <https://doi.org/10.1186/gb-2011-12-5-r44>
35. Pedrós-Alió, C., Manrubia, S.: The vast unknown microbial biosphere. *Proc. Natl. Acad. Sci. U.S.A.* **113**(24), 6585–6587 (2016). <https://doi.org/10.1073/pnas.1606105113>
36. Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K.S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., et al.: A human gut microbial gene catalogue established by metagenomic sequencing. *Nature* **464**(7285), 59–65 (2010)
37. Rodríguez-R, L.M., Konstantinidis, K.T.: Nonpareil: a redundancy-based approach to assess the level of coverage in metagenomic datasets. *Bioinformatics* **30**(5), 629–635 (2014). <https://doi.org/10.1093/bioinformatics/btt584>
38. Roesch, L.F.W., Fulthorpe, R.R., Riva, A., Casella, G., Hadwin, A.K.M., Kent, A.D., Daroub, S.H., Camargo, F.A.O., Farmerie, W.G., Triplett, E.W.: Pyrosequencing enumerates and contrasts soil microbial diversity. *ISME J.* **1**(4), 283–290 (2007). <https://doi.org/10.1038/ismej.2007.53>
39. Rosselli, R., Romoli, O., Vitulo, N., et al.: Direct 16S rRNA-SEQ from bacterial communities: a PCR-independent approach to simultaneously assess microbial diversity and functional activity potential of each taxon. *Sci. Rep.* **6**, 32,165 (2016)
40. Rozov, R., Brown Kav, A., Bogumil, D., Shterzer, N., Halperin, E., Mizrahi, I., Shamir, R.: Recycler: an algorithm for detecting plasmids from de novo assembly graphs. *Bioinformatics* **33**(4), 475–482 (2017)
41. Scher, J.U., Ubeda, C., Artacho, A., et al.: Decreased bacterial diversity characterizes the altered gut microbiota in patients with psoriatic arthritis, resembling dysbiosis in inflammatory bowel disease. *Arthritis Rheumatol.* **67**(1), 128–139 (2015). <https://doi.org/10.1002/art.38892>

42. Schloss, P.D., Girard, R.A., Martin, T., Edwards, J., Thrash, J.C.: Status of the archaeal and bacterial census: an update. *mBio* **7**(3), e00,201-16 (2016). <https://doi.org/10.1128/mBio.00201-16>
43. Schloss, P.D., Handelsman, J.: Status of the microbial census. *Microbiol. Mol. Biol. Rev.* **68**(4), 686–691 (2004). <https://doi.org/10.1128/MMBR.68.4.686-691.2004>
44. Shade, A.: Diversity is the question, not the answer. *ISME J.* **11**(1), 1–6 (2016). <https://doi.org/10.1038/ismej.2016.118>
45. Shakya, M., Quince, C., Campbell, J.H., Yang, Z.K., Schadt, C.W., Podar, M.: Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities. *Environ. Microbiol.* **15**(6), 1882–1899 (2013). <https://doi.org/10.1111/1462-2920.12086>
46. Sharon, I., Kertesz, M., Hug, L.A., et al.: Accurate, multi-kb reads resolve complex populations and detect rare microorganisms. *Genome Res.* **25**(4), 534–543 (2015). <https://doi.org/10.1101/gr.183012.114>
47. Shapton, T.J., Riesenfeld, S.J., Kembel, S.W., et al.: PhyLOTU: a high-throughput procedure quantifies microbial community diversity and resolves novel taxa from metagenomic data. *PLoS Comput. Biol.* **7**(1), e1001,061 (2011)
48. Sogin, M.L., Morrison, H.G., Huber, J.A., et al.: Microbial diversity in the deep sea and the underexplored “rare biosphere”. *Proc. Natl. Acad. Sci. U.S.A.* **103**(32), 12115–12120 (2006). <https://doi.org/10.1073/pnas.0605127103>
49. Sunagawa, S., DeSantis, T.Z., Piceno, Y.M., et al.: Bacterial diversity and White Plague Disease-associated community changes in the Caribbean coral *Montastraea faveolata*. *ISME J.* **3**(5), 512–521 (2009). <https://doi.org/10.1038/ismej.2008.131>
50. Taur, Y., Jenq, R.R., Perales, M.A., et al.: The effects of intestinal tract bacterial diversity on mortality following allogeneic hematopoietic stem cell transplantation. *Blood* **124**, 1174–1182 (2014). <https://doi.org/10.1182/blood-2014-02-554725>
51. Tiedje, J.: Microbial diversity: of value to whom? *ASM News* **60**, 524–525 (1994)
52. Voskoboinik, A., Neff, N.F., Sahoo, D., et al.: The genome sequence of the colonial chordate, *Botryllus schlosseri*. *eLife* **2**, 69 (2013). <https://doi.org/10.7554/eLife.00569>
53. White, R.A., Bottos, E.M., Roy Chowdhury, T., et al.: Moleculo long-read sequencing facilitates assembly and genomic binning from complex soil metagenomes. *mSystems* **1**(3) (2016). <https://doi.org/10.1128/mSystems.00045-16>
54. Williamson, M., Gaston, K.J.: The lognormal distribution is not an appropriate null hypothesis for the species-abundance distribution. *J. Anim. Ecol.* **74**(3), 409–422 (2005). <https://doi.org/10.1111/j.1365-2656.2005.00936.x>
55. Willis, A.: Extrapolating abundance curves has no predictive power for estimating microbial biodiversity. *Proc. Natl. Acad. Sci. U.S.A.* **113**(35), E5096 (2016). <https://doi.org/10.1073/pnas.1608281113>



Chromatyping: Reconstructing Nucleosome Profiles from NOMe Sequencing Data

Shounak Chakraborty^{1,2,3,4}, Stefan Canzar⁴ , Tobias Marschall^{2,3}  ,
and Marcel H. Schulz^{1,2,3}  

¹ Cluster of Excellence for Multimodal Computing and Interaction,
Saarland University, Saarland Informatics Campus E1.7,
66123 Saarbrücken, Germany
mschulz@mmpi.uni-saarland.de

² Max Planck Institute for Informatics, Saarland Informatics Campus E1.4,
66123 Saarbrücken, Germany
t.marschall@mpi-inf.mpg.de

³ Center for Bioinformatics, Saarland University,
Saarland Informatics Campus E2.1, 66123 Saarbrücken, Germany

⁴ Gene Center, Ludwig-Maximilians-Universität München, 81377 Munich, Germany

Abstract. Measuring nucleosome positioning in cells is crucial for the analysis of epigenetic gene regulation. Reconstruction of nucleosome profiles of individual cells or subpopulations of cells remains challenging because most genome-wide assays measure nucleosome positioning and DNA accessibility for thousands of cells using bulk sequencing. Here we use characteristics of the NOMe-sequencing assay to derive a new approach, called ChromaClique, for deconvolution of different nucleosome profiles (chromatypes) from cell subpopulations of one NOMe-seq measurement. ChromaClique uses a maximal clique enumeration algorithm on a newly defined NOMe read graph that is able to group reads according to their nucleosome profiles. We show that the edge probabilities of that graph can be efficiently computed using Hidden Markov Models. We demonstrate using simulated data that ChromaClique is more accurate than a related method and scales favorably, allowing genome-wide analyses of chromatypes in cell subpopulations. Software is available at <https://github.com/shounak1990/ChromaClique> under MIT license.

Keywords: NOMe-seq · Max clique enumeration · Epigenetics
HMMs

1 Introduction

The eukaryotic genome is organized in nucleosomes which consist of approximately 147 base pairs of DNA wrapped around a histone octamer. Nucleosomes serve as the basic unit of chromatin packaging and are connected via free DNA

linkers of variable length. Nucleosome positioning plays a pivotal role for transcriptional regulation by controlling DNA accessibility for binding proteins (*e.g.* transcription factors). Thus, learning more about nucleosome positioning and how it differs between different cell types, as well as subpopulations of cells, is an important task to understand gene expression regulation.

Different protocols for the genome-wide characterization of nucleosome positioning have been developed. The most common are DNaseI-seq [1], ATAC-seq [2] and NOME-seq [3]. NOME-seq (nucleosome occupancy and methylation) utilizes the enzyme M.CviPI which specifically methylates cytosine dyads in a GpC sequence context. Because NOME-seq uses bisulfite sequencing, it also delivers the endogenous CpG methylation levels, enabling the simultaneous analysis of chromatin accessibility and DNA methylation. Due to this unique feature, a number of recent studies have applied NOME-seq to study epigenetic regulation [3–7]. It is also the first assay that can measure nucleosome positioning and DNA methylation simultaneously in single cells [8].

However, single cell datasets using NOME-seq or other related assays are rare, whereas bulk sequencing experiments do not reveal nucleosome and chromatin profiles of subpopulations of cells. Although NOME-seq is normally obtained from bulk sequencing of cells, the nucleosome readout of one paired-end read comes from a single cell. As several GpC dinucleotides may appear on a paired-end read obtained from NOME-seq, this information can be used to group reads that originate from the same nucleosome profile. We call these distinct nucleosome profiles *chromatypes*, to emphasize that their chromatin arrangement differs between cells. Here, we are concerned with the development of novel computational methods that can reconstruct chromatypes from NOME-seq data.

The only comparable method is epiG, which clusters reads according to epigenetic haplotypes using a Bayesian approach that considers DNA methylation and GpC methylation in NOME-seq data [9]. However, the Bayesian approach in epiG is slow and can thus only be used to study local genomic regions and does not allow genome-wide application.

We exploit recent advances for methods that reconstruct viral haplotypes from DNA-seq data. The high mutation rates of viruses such as HIV give rise to considerable intra-patient variability of virus genomes [10]. Reconstructing the full set of virus haplotypes circulating in a patient’s blood and quantifying their relative abundances are important tasks with the prospect of informing therapy stratification [11]. This computational task is challenging, however, because usually no *a priori* knowledge on the number of haplotypes and the distribution of their abundances is available. Therefore, distinguishing sequencing errors from low-abundance haplotypes requires non-trivial techniques. In the meantime, a wealth of methods has been developed [12], including HaploClique that enumerates maximal cliques on a DNA-seq read graph [13].

We introduce a novel method, called ChromaClique, which combines the maximal-clique enumeration procedure of HaploClique with a novel probabilistic edge criterion tailored to NOME-seq data. The edge criterion incorporates base quality scores in a probabilistic manner. ChromaClique uses Hidden Markov Models for the efficient computation of the edge probabilities in the novel read

graph. We show that ChromaClique is the first algorithm that can be used genome-wide and that it has better accuracy on simulated data compared to the only comparable method epiG.

2 Methods

2.1 ChromaClique Overview

ChromaClique starts from bulk NOME-seq reads aligned to a reference genome in BAM format. Each cell, or group of cells, is expected to have different nucleosome positioning patterns (chromatypes) which are encoded in the reads. This is depicted in Fig. 1 with the different colors, where each color represents a chromatype. The aligned reads are converted into a *read graph*, $G := (V, E)$, with nodes V and edges E . Each node represents a read. Two reads share an edge only if they are likely to originate from the same chromatype. Both single and paired-end reads are considered for the edge criterion. Two paired-end reads share an edge only when both reads from both pairs agree to the edge criterion. The maximal cliques in the graph are enumerated using the algorithm previously employed in HaploClique [13]. The reads in a maximal clique are merged. The condensed graph is checked again for cliques which have an edge between each other and the maximal clique finding algorithm is run iteratively. This continues until no more edges are found in the graph. The nodes in the final graph represent the individual reconstructed chromatypes and are also called super reads.

2.2 Encoding the Reads

In NOME-seq data only GCH trinucleotides, *i.e.* GCT, GCA or GCC, in the genome provide information about open and closed nucleosome positions,

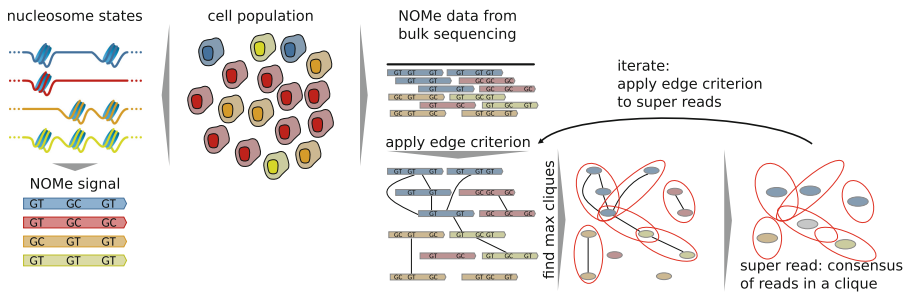


Fig. 1. Illustration of a cell population with different nucleosome states, indicated by different colors. The NOME signature of different chromatin states is shown on the bottom left. The ChromaClique workflow is shown on the right: ChromaClique applies its edge criterion to NOME bulk sequencing data (black lines connecting reads), enumerates all maximal cliques (indicated in red), merges reads in a clique, and iterates the process until convergence. (Color figure online)

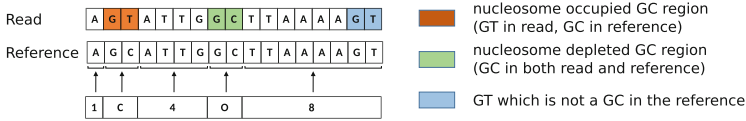


Fig. 2. GCH and GTH positions in a read are compared with the reference. If there is a position with GCH in the read and a GCH in the reference, it is marked as “O” and otherwise if it is a GTH in the read it is marked as “C”. All other positions other than the GCH or GTH positions are summarized by numbers reflecting their length. If there is a GTH in the read and a GTH in the reference it is not treated as a GCH position.

because GCG positions are ambiguous due to the possibility that CpG DNA methylation took place. Each individual read is represented as follows: each GCH position is encoded as open (O) and each GTH position is encoded as closed (C). This is because a GCH is not converted by bisulfite treatment, when it is accessible and was methylated by the enzyme. The NOME enzyme M.cviPI works on accessible GCs on both DNA strands in 5’ to 3’ direction. This reverse complementarity is taken into account during read encoding and edge construction by the algorithm. Figure 2 shows the process of encoding a read based on its GCH and GTH occurrences. The last GTH position is not a closed position since the reference does not have a GCH position there.

Sequencing errors in the reads that would prevent the detection of GCH can be corrected by comparing the positions in the reference sequence to which the read is aligned. For example if there is a GCH position in the reference and due to an error the read contains HCH instead this will be corrected for later use in the algorithm.

2.3 Definition of Edge Probabilities

In order to build the graph for finding maximum cliques, each pair of reads with sufficient overlap is scored against each other to see if they are likely to originate from cells with the same chromatype. The reads are scored on the basis of their base quality scores (Phred scores) as reported by the sequencer, and also based on the similarity of nucleotides observed at their shared GCH positions.

ChromaClique does not make an assumption on the number and relative abundances of open-chromatin patterns. In order to evaluate the likelihood of an edge between two considered reads, we compute the *edge probability* as the probability that the overlapping portion of both reads has been generated by any one of the possible chromatypes. Before we can properly define the edge probability we make a number of definitions.

ChromaClique first encodes reads at only GCH positions (in the following called C/O positions) and records the distance between consecutive occurrences (see Fig. 2). In the read each GC is denoted as open (O) and each GT as closed (C). For simplicity, we denote $C_i(R)$ as the open-chromatin status (O or C) at the i_{th} C/O position in read R , e.g., $C_2(R) = O$ in Fig. 2. The phred base

quality of the Cytosine or Thymine at the i_{th} C/O position in read R is denoted $phred(i, R)$. Let $Q_i(R)$ be the scaled base quality score at position i , that is $Q_i(R) = 10^{-\frac{phred(i, R)}{10}}$. The distance between the i_{th} and j_{th} C/O position in the read is given by $d_{i,j}(R)$, e.g., $d_{1,2}(R) = 4$ in Fig. 2.

Computing the edge probability involves two steps. The first estimates the probability for a given chromatype y given the base qualities obtained from the sequencer, denoted $P(R|y)$. Let T be the total number of C/O positions in an encoded read, then:

$$P(R|y) = \prod_{i=1}^T f_{qual}(R, y, i), \quad (1)$$

where f_{qual} is defined as:

$$f_{qual}(R, y, i) = \begin{cases} 1 - Q_i(R), & \text{if } C_i(R) = C_i(y) \\ Q_i(R), & \text{if } C_i(R) \neq C_i(y) \end{cases}. \quad (2)$$

The second step consists in computing the probability of an individual chromatype y , denoted as $P(y)$. A nucleosome occupies around 147 bps and therefore not all possible chromatypes are equally likely. For example 1C 2C 2C 2C is more likely than 1C 2O 2C 2O. We capture this by defining transition events at adjacent C/O positions.

For a read R a *transition* for position i is defined as $C_i(R) \neq C_{i+1}(R)$, namely the open-chromatin state at position i has changed compared to its adjacent position $i+1$. Here we do not distinguish the direction of the transition, i.e. a transition from an O to a C is equivalent to a transition of a C to an O. Similarly, position i is called a *non-transition* if $C_i(R) = C_{i+1}(R)$. As mentioned above, the distance d between two positions i and j should influence the likelihood of a transition event. Therefore we obtain the empirical transition probability $tr(d)$, as the relative frequency of transition events for a certain distance d :

$$tr(d) = \frac{Transition(d)}{Transition(d) + NonTransition(d)}, \quad (3)$$

where $Transition(d)$ and $NonTransition(d)$ are the number of transition and non-transition events at distance d observed in all reads, respectively. Then the non-transition probability is simply given by:

$$1 - tr(d). \quad (4)$$

Transition or non-transition probabilities are used in the computation of observing a certain C/O pattern in a read. In addition, these probabilities may help to recognize errors in the reads, for instance errors due to the incorrect methylation of the M.CviPI enzyme, or due to incorrect bisulfite conversion. For example if the transition probability for a specific distance, say 10, is 0.05, it means that the number of non-transitions seen for this distance is much higher than the number of transitions. However, if a transition was observed at this distance, the probability that it is an error due to either a failed NOME or

bisulfite conversion, would be high. This information is later used as a prior when two reads are compared to see if they originate from cells with similar chromatypes.

Finally, we can use the transition probabilities (Eq. 3) to quantify the probability of observing a particular chromatype y . We define:

$$P(y) = \prod_{i=1}^{T-1} f_{transition}(y, i), \quad (5)$$

$$f_{transition}(y, i) = \begin{cases} 1 - tr(d_{i-1,i}(y)), & \text{if } C_i(y) = C_{i-1}(y) \text{ and } i > 1 \\ tr(d_{i-1,i}(y)) & \text{if } C_i(y) \neq C_{i-1}(y) \text{ and } i > 1. \\ 0.5 & i = 1 \end{cases} \quad (6)$$

Intuitively, $P(y)$ will be low if the chromatin state configuration in y is unlikely given the transition probabilities. If two reads R_1 and R_2 are independent of each other, the probability that they originate from a particular chromatype y can now be calculated as follows:

$$P(R_1, R_2|y) = P(R_1|y) P(R_2|y). \quad (7)$$

From the law of total probability, the probability that two reads originate from the same chromatype can be computed as:

$$P(R_1, R_2) = \sum_{y \in Y} P(R_1, R_2|y) P(y), \quad (8)$$

where Y is the set of all possible 2^T chromatypes. Equation (8) is the central *edge probability* of ChromaClique that is used for building its read graph. Two reads are said to be from the same chromatype if the probability $P(R_1, R_2)$ is above a threshold δ . We call δ the *edge threshold* and only edges with $P(R_1, R_2) > \delta$ are considered in the read graph. δ needs to be set manually by the user, but we will determine a practical value for δ using simulations.

Minimum Overlap. The edge probability depends on another parameter which also needs to be set manually. It is the number of C/O positions, D , in the overlapping portion of the two reads in question. If D is too small then this may lead to false edges between reads originating from different chromatypes. However if the number is too large then it leads to many read overlaps not being considered. By default we set the minimum number of overlapping C/O positions to 2.

Thus this parameter determines the purity of the cliques and also the length of the final super reads. It was set manually after analysing the behaviour of simulated data.

2.4 Efficient Calculation of Edge Probabilities in ChromaClique

The probability that two reads originate from the same chromatype is given by Eq. (8). In order to obtain the above probability R_1 and R_2 have to be

checked against all the possible chromatypes, i.e the entire set Y . The size of Y is 2^T , where T is the number of C/O positions in the overlapping portion of the reads. Thus, it becomes computationally expensive to enumerate all the different chromatypes and then calculate the probability.

However, if the overlapping portion of the reads is modeled as a Hidden Markov Model (HMM), the forward algorithm can be used to efficiently calculate the entire probability without having to enumerate all the possible chromatypes.

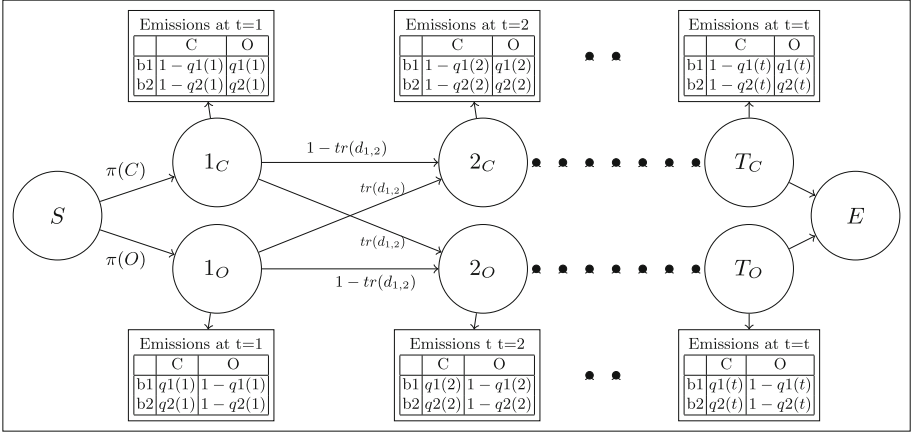


Fig. 3. Hidden Markov Model for calculating the probability that two reads originate from the same chromatype. The circles (1_C , 1_O , 2_C ...) represent the hidden states which are the actual open or closed state of the chromatin in the DNA sequence (based on GCH or GTH positions). Each of these states emits two values (one for each of the sequences being compared). The emission probabilities for these are given by the tables near these states. The transition probabilities from one hidden state to another is given by the arrows between the hidden states. The start state S and end state E are customary states denoting the start and end of the process.

Hidden Markov Model (HMM) for Chromatyping. Figure 3 illustrates the HMM for calculating the probability that two reads originate from the same chromatype. It consists of a set of hidden states, which represent the actual nucleotide state (open or closed). Each hidden state emits a pair of nucleotides, one nucleotide for each read at a C/O position. The emission parameters consider the phred base qualities.

More formally, let T be the total number of C/O positions in the overlapping region of the two reads (R_1 and R_2) being compared. Let $t \in \{1, \dots, T\}$ be the index for the C/O positions, where $R_1(t) \in \{C, O\}$ and $R_2(t) \in \{C, O\}$ denote the chromatin status given by R_1 and R_2 at position t , respectively. Let $\{S, 1_C, 1_O, 2_C, 2_O, \dots, T_C, T_O, E\}$ represent the set of hidden states, where S and E denote the silent *start* and *end* state, respectively. In the following we will refer to a state from the set as t_b with $t \in \{1, \dots, T\}$ and $b \in \{C, O\}$.

State t_b has emission probability $e_{t_b}(b_1, b_2)$ for a pair (b_1, b_2) , with $b_i \in \{C, O\}$, defined as:

$$e_{t_b}(b_1, b_2) = \begin{cases} (1 - q_1(t)) \cdot (1 - q_2(t)) & b = b_1 \text{ and } b = b_2, \\ q_1(t) \cdot (1 - q_2(t)) & b \neq b_1 \text{ and } b = b_2, \\ (1 - q_1(t)) \cdot q_2(t) & b = b_1 \text{ and } b \neq b_2, \\ q_1(t) \cdot q_2(t) & b \neq b_1 \text{ and } b \neq b_2, \end{cases} \quad (9)$$

where $q_1(t)$ is defined as:

$$q_1(t) = 10^{-\frac{\text{phred}(t, R_1)}{10}} \quad (10)$$

and $q_2(t)$ is defined analogously for R_2 .

The initial probabilities from the start state S to 1_C and 1_O are set to $\pi(C) = 0.5$ and $\pi(O) = 0.5$, respectively. The transition probabilities between consecutive states $(t-1)_b$ and t_c , with $b, c \in \{C, O\}$, are defined using the transition probability $tr(d)$ for distance d between C/O positions $t-1$ and t :

$$a_{(t-1)_b, t_c} = \begin{cases} 1 - tr(d_{t-1, t}) & b = c, \\ tr(d_{t-1, t}) & b \neq c. \end{cases} \quad (11)$$

We can now compute the sought probability $P(R_1, R_2)$, Eq. (8), using the standard forward algorithm for HMMs [14]. The complexity of calculating the probability of two reads originating from one chromatype using the forward algorithm is $\mathcal{O}(T)$, where T is the number of C/O positions in the overlapping portion of the reads.

3 Data Simulation and Evaluation

To assess performance with respect to ground truth chromatypes, which are usually not available for real data, we simulated NOME sequencing experiments *in silico*. Simulated data also serve to tune parameters as needed, in particular δ , the threshold for the probability that two reads originate from cells with the same chromatypes, and \mathbf{D} , the minimum number of C/O positions we require in the overlapping region of two reads.

3.1 Simulating Chromatypes

The reference sequence of human chromosome 1 was randomly annotated with regions of open chromatin and closed chromatin. Regions of 177 bps were annotated with a nucleosome (closed chromatin for 147 bps) followed by a linker DNA (open chromatin for 30 bps) with a 60% probability. The whole region (177 bps) was annotated as being open chromatin with a 40% probability. This process of annotation was done along the complete chromosome 1. The process was repeated four times in order to simulate four different chromatypes.

Virtual NOME and bisulfite treatment was simulated as follows: GCHs in nucleosome occupied regions were converted to GTHs. In regions not occupied

by nucleosomes and in linker DNA regions, GCHs were retained. We randomly methylated HCGs, i.e., sites of DNA methylation. In this way each chromatype had distinct open chromatin (GCHs) and DNA methylation (HCGs) profiles, where DNA methylation values are currently only used by epiG.

3.2 Simulating NGS Reads

Illumina sequencing reads were simulated (along with sequencing errors), individually for each of the simulated chromatypes using the ART software [15] and subsequently merged using samtools. The merged reads were aligned to the reference using BISMARK [16]. Four different sets of merged reads, 100 bp reads at 40× and 80× coverage, as well as 200 bp reads at 40× and 80× coverage, were created. We chose 100 bp reads since this is a common read length, while 200 bp reads were included to evaluate the impact of read length on performance. ChromaClique and epiG were run individually on each of these datasets.

3.3 Evaluation Metric for Chromatyping Reconstructions

The chromatyping reconstructions produced by the algorithms were evaluated based on the number of switches needed to reconstruct that particular super read from the four ground truth simulated chromatypes. Each super read (chromatyping-reconstruction) was represented by a binary vector, $Sr[x]$, containing 1s and 0s, for open and closed positions, respectively.

For example, let $S = 10\ 42C\ 23C\ 9C$ be a reconstructed super read. This super read can be represented as a binary vector Sr containing 1s and 0s for open and closed positions respectively, $Sr = [1, 0, 0, 0]$.

Because the super reads are aligned to the reference, similar vectors can be constructed for each of the ground truth chromatypes that were used for simulating the data. This produces a chromatyping matrix $Chr[c, x]$, where each row c represents one of the ground truth chromatypes and each column x represents the nucleosome state (1 or 0) at that position.

For example assume the following chromatyping matrix Chr :

$$Chr[c, x] = \begin{matrix} chromatyping1 \\ chromatyping2 \\ chromatyping3 \\ chromatyping4 \end{matrix} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \quad (12)$$

The number of switches (jumps from one original chromatyping to another) required to recreate a particular super read (read group in case of epiG), is referred to as the *switch error*. SE_i is the switch error for super read i . The switch error can be efficiently calculated from the $Sr[x]$ vector and the $Chr[c, x]$ matrix using dynamic programming.

With $SE[c, x]$ we denote the *switch error matrix*, where a row c represents one of the initial chromatypes and an entry in column x denotes the minimum number of switches and mismatches needed to reconstruct a prefix of length x in

Evaluation of the Output from ChromaClique. ChromaClique outputs a BAM file containing both paired-end and single-end super reads, which are aligned to the reference. Each super read represents local reconstructions of a chromatype. For a single-end super read, the *Chr* matrix and *Sr* vectors can be directly constructed from the nucleotide positions (open or closed), in the super read and the initial chromatypes used for simulation. Thus, the switch error can be calculated directly.

However, for paired-end reads, there is missing information in between the two read ends and the *Chr* matrix needs to be constructed for an individual pair. Essentially, only positions that are overlapped by one of the reads in the super read pair are part of the corresponding *Chr* matrix for that super read, ignoring C/O positions in the reference that are not overlapped by the super read.

Evaluation of the Output from epiG. The output from epiG is not exactly the same as that from ChromaClique. While ChromaClique reports reconstructed local chromatypes obtained by merging reads from the initial aligned reads, epiG assigns reads to “epigenetic haplotypes” [9]. In order to compare the outputs of both algorithms, the overlapping reads of epiG were merged using the same algorithm that is used to merge the reads in ChromaClique. The switch errors and prediction error for epiG were calculated using these merged reads as explained above.

BaseLine Chromatype. In order to assess the performance of the algorithms ChromaClique and epiG, a *BaseLine chromatype* was constructed, which was composed of only closed positions. The idea of the BaseLine is to measure the error for the simplest possible predictor. The switch error and prediction error were calculated for the BaseLine chromatype in the same way. The percentage of coverage was varied for the BaseLine chromatype to simulate insufficient coverage scenarios.

4 Results

We generated simulated data for the evaluation of the epiG and ChromaClique algorithms. First, we compared the relationship between transition rates and distances between our simulated data and real HepG2 NOME sequencing data (Fig. 4). As expected, the probability of observing a transition goes up as the distance increases between two consecutive GCH occurrences and plateaus at a certain value. This general trend is observed for both the real and simulated data.

We then compared the performance of ChromaClique on the simulated datasets to epiG [9], which was run in “NOME-seq” mode with the minimum number of GCH positions (`min_DGCH` flag) set to 2. The way epiG outputs chromatypes is different from ChromaClique and therefore some post-processing was

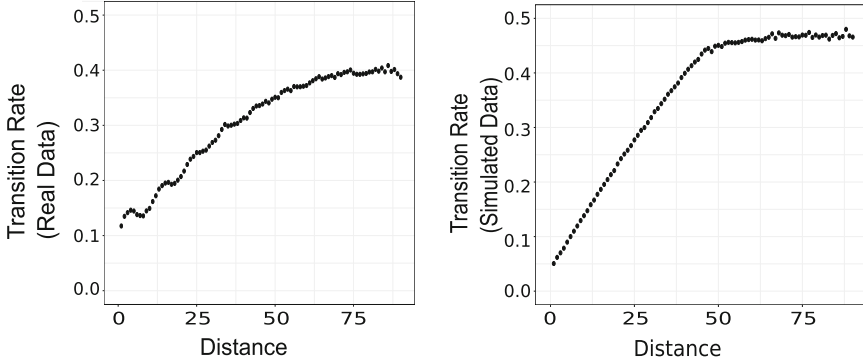


Fig. 4. Plots showing the transition rates at different distances between consecutive GCH occurrences for forward strand reads mapped to chromosome 1 for HepG2 data (left) and simulated data (right).

required to compare the two algorithms. epiG assigns each read to an epigenetic haplotype (comparable to a chromatype). All reads belonging to a particular epigenetic haplotype were merged (in overlapping regions), and this was considered as a reconstruction of a chromatype. Merging the overlapping reads was done using the same merging algorithm as in ChromaClique. Each merged read group from epiG was evaluated in the same way as each super read reported by ChromaClique. The performance of a *BaseLine chromatype* containing only closed positions over the length of the considered region was evaluated as a control for the performances of ChromaClique and epiG.

The evaluation was done using the prediction error, which denotes the average number of switch errors obtained for all predicted super reads of a method (see subsection 3.3). Another criterion for evaluation of the performance of the different algorithms is the fraction of C/O positions in the original genomic region that was covered by the reconstructed chromatypes. In this way, we can assess the trade-off between a low switch error rate and a high fraction of C/O positions covered. The threshold parameter δ in ChromaClique allows to adjust this trade-off, whereas there are no such parameters in epiG. The evaluation was restricted to a region of size 100000 bps, because epiG could not be run on the whole chromosome 1, see below.

Figure 5 shows the prediction errors of ChromaClique (green triangles) for thresholds varying from 0.000001 to 0.45, plotted against the fraction of C/O positions that were covered by the predictions. Decreasing values of δ lead to a higher fraction of GC regions being covered in the output while the errors remain constant for a certain range of thresholds. Above a certain threshold, the errors increase steadily. This behavior is noticed for all four different simulated datasets. The least prediction errors were reported for the thresholds of 0.05 and 0.07 for 100 bp and 200 bp reads, respectively.

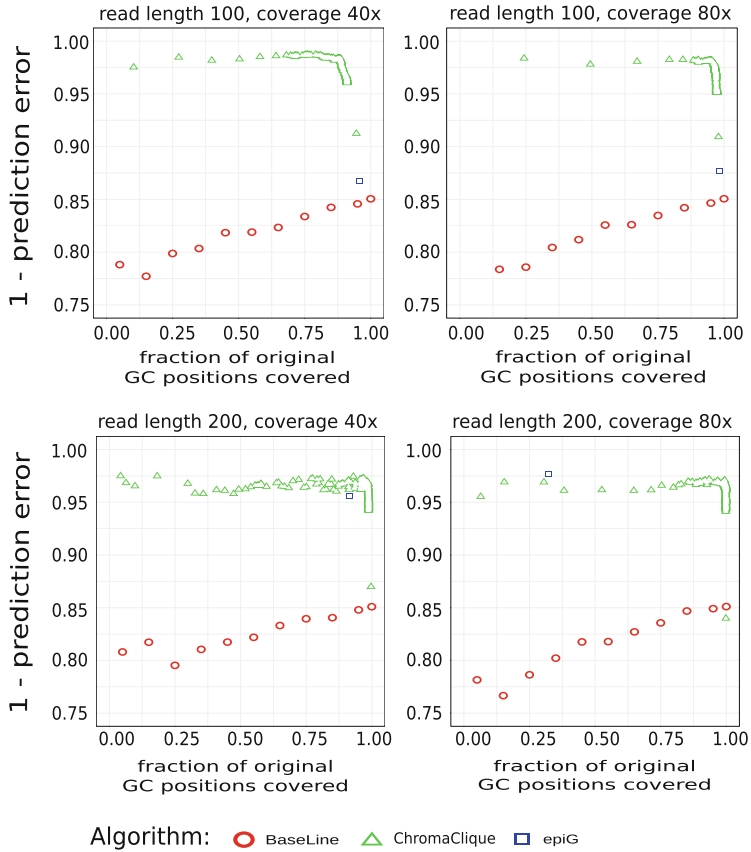


Fig. 5. Plots comparing the performance of ChromaClique with that of epiG and also a BaseLine chromatyping reconstruction for four simulated data sets with different read lengths (100 or 200) and coverages (40x and 80x). (Color figure online)

We sampled varying percentages of the original GC positions to be covered by the BaseLine chromatyping. In this way, we mimicked different trade-offs between error rate and fraction of covered positions, as shown by red circles in Fig. 5. For all data sets, we noticed a trend towards higher prediction error rates when fewer GC positions are covered. We observed that the number of switch errors decreases at a smaller rate than the number of GC positions covered and therefore the prediction error increases.

Figure 5 also shows the performance of epiG. Since epiG provides no parameter with which it can be tuned to get varying performances, only one error value could be obtained for each simulated dataset (blue square). For the 100 bp reads, the fraction of C/O positions covered by epiG is high at the cost of relatively high error rates, which are hardly better than the BaseLine chromatyping. It seems to profit in terms of the prediction error with an increase in the length

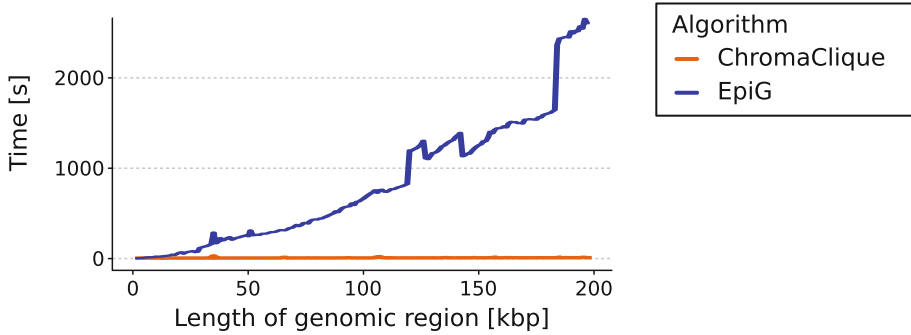


Fig. 6. Runtime of ChromaClique and epiG as a function of the length of the processed region for 100 bp reads and 40x coverage.

of the reads to 200 bps, and yields error rates which are similar to those of ChromaClique. However, an unexpected drop in C/O position coverage is noticed for the dataset with 200 bp reads and 80 \times coverage.

Figure 6 shows the runtimes of ChromaClique and epiG plotted against the size of the genomic region from which the initial aligned reads were sequenced. While ChromaClique’s runtime grows slowly (and appears almost constant at the scale shown in Fig. 6), the runtime of epiG increases steadily with growing region sizes. While ChromaClique can be run on a chromosome-wide scale (≈ 101 min for the entire human chromosome 1 on 100 bp and 40X coverage data), the runtime for epiG becomes prohibitively large for regions more than 1 million base pairs.

5 Discussion and Conclusion

In this paper, we introduced ChromaClique, a novel algorithm to reconstruct nucleosome profiles from NOMe-seq data. ChromaClique is the first tool that scales to whole genomes. Furthermore, it outperforms epiG, the only competitor, in terms of prediction error rates and prediction completeness.

ChromaClique comes with the advantage that it only considers read pairs that have a sufficient C/O position overlap and then predicts whether the overlapping reads originate from the same chromatype. In contrast, epiG takes all provided reads and decides which chromatype a read is to be assigned to based on a likelihood score. That is, epiG assigns every read to a chromatype, but does not output information on where the chromatype reconstructions are reliable.

We note that NOMe-seq provides information about open and closed nucleosome positions based on the GCH regions. It hence comes with the intrinsic limitation of not being able to provide any information in GCH deserts. Thus, the reconstruction of nucleosome profiles is not possible in regions of low GC density using this protocol and we consider extending ChromaClique to accommodate other data types a fruitful direction for future research.

The runtime of ChromaClique depends on the number of cliques in the NOME read graph, where an edge between two reads is defined by read overlaps. The number of cliques can potentially increase exponentially with an increase in the coverage. For constant coverage, however, ChromaClique scales linearly with the length of the considered region (in practice). epiG takes a different approach in its optimization algorithm. Starting from all reads as singletons initially, it optimizes for chains of reads that are overlapping each other using a likelihood formulation that uses priors on preferred lengths of read chains to search through the large space of possible combinatorial configurations. Thus, the optimization algorithm in epiG depends on the initial size of the region selected, as non-overlapping reads are considered to be part of the same haplotype chain throughout the algorithm. Our experiments suggest that for moderate to high coverage values, the speed of ChromaClique is sufficient and scales much better than the approach taken in epiG.

ChromaClique has shown a consistent performance across the different simulated datasets in terms of prediction error and the length of C/O positions covered. It consistently achieves lower error rates than epiG with the 100 bp reads. For the 200 bp reads, epiG shows similar error values to ChromaClique but lower coverage of C/O positions for the 80x case. One of the advantages of ChromaClique over epiG is its ability to tune the performance using the threshold parameter. This allows users to employ different thresholds for different datasets. For our experiments with simulated data, the thresholds that were most effective were between 0.05 to 0.07.

ChromaClique is a new method which allows for the reconstruction and subsequent analysis of nucleosome profiles on a chromosome-wide scale. In future work, it would be interesting to improve the simple simulation strategy by designing a more realistic simulation scenario, by combining real NOME-seq data sets of different conditions. It would also be interesting to extend the model to consider DNA methylation at CpG residues as well. A promising application domain of ChromaClique is single cell NOME-seq data, which we plan to explore in the future.

Acknowledgments. We thank Karl Nordström, Gilles Gasparoni and Jörn Walter for providing access to the HepG2 NOME-seq data.

References

1. Thurman, R.E., et al.: The accessible chromatin landscape of the human genome. *Nature* **489**(7414), 75–82 (2012)
2. Buenrostro, J.D., Giresi, P.G., Zaba, L.C., Chang, H.Y., Greenleaf, W.J.: Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat. Methods* **10**(12), 1213–1218 (2013)
3. Kelly, T.K., Liu, Y., Lay, F.D., Liang, G., Berman, B.P., Jones, P.A.: Genome-wide mapping of nucleosome positioning and DNA methylation within individual DNA molecules. *Genome Res.* **22**(12), 2497–2506 (2012)

4. Taberlay, P.C., Statham, A.L., Kelly, T.K., Clark, S.J., Jones, P.A.: Reconfiguration of nucleosome-depleted regions at distal regulatory elements accompanies DNA methylation of enhancers and insulators in cancer. *Genome Res.* **24**(9), 1421–1432 (2014)
5. Durek, P., et al.: Epigenomic profiling of human CD4⁺ T cells supports a linear differentiation model and highlights molecular regulators of memory development. *Immunity* **45**(5), 1148–1161 (2016)
6. Guo, H., et al.: DNA methylation and chromatin accessibility profiling of mouse and human fetal germ cells. *Cell Res.* **27**(2), 165–183 (2017)
7. Schmidt, F., et al.: Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction. *Nucleic Acids Res.* **45**(1), 54–66 (2017)
8. Pott, S.: Simultaneous measurement of chromatin accessibility, DNA methylation, and nucleosome phasing in single cells. *eLife* **6**, e23203 (2017)
9. Vincent, M., et al.: epiG: statistical inference and profiling of DNA methylation from whole-genome bisulfite sequencing data. *Genome Biol.* **18**(1), 38 (2017)
10. Domingo, E., Sheldon, J., Perales, C.: Viral quasispecies evolution. *Microbiol. Mol. Biol. Rev.* **76**(2), 159–216 (2012)
11. Beerenwinkel, N., Günthard, H.F., Roth, V., Metzner, K.J.: Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Front. Microbiol.* **3**, 329 (2012)
12. Posada-Céspedes, S., Seifert, D., Beerenwinkel, N.: Recent advances in inferring viral diversity from high-throughput sequencing data. *Virus Res.* **239**, 17–32 (2017)
13. Töpfer, A., Marschall, T., Bull, R.A., Luciani, F., Schönhuth, A., Beerenwinkel, N.: Viral quasispecies assembly via maximal clique enumeration. *PLoS Comput. Biol.* **10**(3), e1003515 (2014)
14. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**(2), 257–286 (1989)
15. Huang, W., Li, L., Myers, J.R., Marth, G.T.: ART: a next-generation sequencing read simulator. *Bioinformatics* **28**(4), 593–594 (2012)
16. Krueger, F., Andrews, S.R.: Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics* **27**(11), 1571–1572 (2011)



GTED: Graph Traversal Edit Distance

Ali Ebrahimpour Boroojeny¹, Akash Shrestha¹, Ali Sharifi-Zarchi^{1,2,3},
Suzanne Renick Gallagher¹, S. Cenk Sahinalp⁴, and Hamidreza Chitsaz¹✉

¹ Colorado State University, Fort Collins, CO, USA

chitsaz@chitsazlab.org

² Royan Institute, Tehran, Iran

³ Sharif University of Technology, Tehran, Iran

⁴ Indiana University, Bloomington, IN, USA

<http://chitsazlab.org>

Abstract. Many problems in applied machine learning deal with graphs (also called networks), including social networks, security, web data mining, protein function prediction, and genome informatics. The kernel paradigm beautifully decouples the learning algorithm from the underlying geometric space, which renders graph kernels important for the aforementioned applications.

In this paper, we give a new graph kernel which we call graph traversal edit distance (GTED). We introduce the GTED problem and give the first polynomial time algorithm for it. Informally, the graph traversal edit distance is the minimum edit distance between two strings formed by the edge labels of respective Eulerian traversals of the two graphs. Also, GTED is motivated by and provides the first mathematical formalism for sequence co-assembly and *de novo* variation detection in bioinformatics.

We demonstrate that GTED admits a polynomial time algorithm using a linear program in the graph product space that is guaranteed to yield an integer solution. To the best of our knowledge, this is the first approach to this problem. We also give a linear programming relaxation algorithm for a lower bound on GTED. We use GTED as a graph kernel and evaluate it by computing the accuracy of an SVM classifier on a few datasets in the literature. Our results suggest that our kernel outperforms many of the common graph kernels in the tested datasets. As a second set of experiments, we successfully cluster viral genomes using GTED on their assembly graphs obtained from *de novo* assembly of next generation sequencing reads. Our GTED implementation can be downloaded from <http://chitsazlab.org/software/gted/>.

1 Introduction

Networks, or graphs as they are called in mathematics, have become a common tool in modern biology. Biological information from DNA sequences to protein interaction to metabolic data to the shapes of important biological chemicals are often encoded in networks.

One goal in studying these networks is to compare them. We might want to know whether two DNA assembly graphs produce the same final sequences

or how close the protein interaction networks of two related species are. Such comparisons are difficult owing to the fact that determining whether two graphs have an identical structure with different labels or vertex ordering is an NP-complete problem. Therefore, any comparisons will need to focus on specific aspects of the graph.

Here, we present the notion of *graph traversal edit distance (GTED)*, a new method of comparing two networks. Informally, GTED gives a measure of similarity between two directed Eulerian graphs with labeled edges by looking at the smallest edit distance that can be obtained between strings from each graph via an Eulerian traversal. GTED was inspired by the problem of *differential genome assembly*, determining if two DNA assembly graphs will assemble to the same string. In the differential genome assembly problem, we have the de Bruijn graph representations of two (highly) related genome sequence data sets, where each edge e represents a substring of size k from *reads* extracted from these genome sequences (e.g. one from a cancer tissue and the other from the normal tissue of the same individual), and its multiplicity represents the number of times its associated substring is observed in the reads of the respective genome sequence. In this formulation, each vertex represents the $k - 1$ length prefix of the label of its outgoing edges and the $k - 1$ length suffix of the label of its incoming edges. Thus, the labels of all incoming edges of a vertex (respectively all outgoing edges) are identical with the exception of their first (last) symbol. Differential genome assembly has been introduced to bioinformatics in two flavors: (i) *reference genome free* version [1–5], and (ii) *reference genome dependent* version, which, in its most general form, is NP-hard [6]. Both versions of the problem are attracting significant attention in biomedical applications (e.g. [7, 8]) due to the reduced cost of genome sequencing (now approaching \$1000 per genome sample) and the increasing needs of cancer genomics where tumor genome sequences may significantly differ from the normal genome sequence from the same individual through single symbol edits (insertions, deletions and substitutions) as well as block edits (duplications, deletions, translocations and reversals).

In addition to comparing assembly graphs, GTED can also be used to compare other types of networks. GTED yields a (pseudo-)metric for general graphs because it is based on the edit distance metric. Hence, it can be used as a graph kernel for a number of classification problems. GTED is the first mathematical formalism in which global traversals play a direct role in the graph metric. In this paper, we give a polynomial time algorithm using linear programming that is guaranteed to yield an integer solution. We use that as a graph kernel, and evaluate the performance of our new kernel in SVM classification over a few datasets. We also use GTED for clustering of viral genomes obtained from *de novo* assembly of next generation sequencing reads. Note that GTED is a global alignment scheme that is not immediately scalable to full-size large genomes, like all other global alignment schemes such as Needleman-Wunsch. However, GTED can form the mathematical basis for scalable heuristic comparison of full-size large genomes in the future.

Related Work

Many problems in applied machine learning deal with graphs, ranging from web data mining [9] to protein function prediction [10]. Some important application domains are biological networks such as regulatory networks, sequence assembly and variation detection, and structural biology and chemoinformatics where graphs capture structural information of macromolecules. For instance, machine learning algorithms are often used to screen candidate drug compounds for safety and efficacy against specific diseases and also for repurposing of existing drugs [11]. Kernel methods elegantly decouple data representation from the learning part; hence, graph learning problems have been studied in the kernel paradigm [12]. Following [12], other graph kernels have been proposed in the literature [13].

A graph kernel $k(G_1, G_2)$ is a (pseudo-)metric in the space of graphs. A kernel captures a notion of similarity between G_1 and G_2 . For instance for social networks, k may capture similarity between their clustering structures, degree distribution, etc. For molecules, similarity between their sequential/functional domains and their relative arrangements is important. A kernel is usually computed from the adjacency matrices of the two graphs, but it must be invariant to the ordering (permutation) of the vertices. That property has been central in the graph kernels literature.

Existing graph kernels that are vertex permutation invariant use either local invariants, such as counting the number of triangles, squares, etc. that appear in G as subgraphs, or spectral invariants captures as functions of the eigenvalues of the adjacency matrix or the graph Laplacian. Essentially, different graph kernels ranging from random walks [12] to shortest paths [14, 15] to Fourier transforms on the symmetric group [16] to multiscale Laplacian [17] compute local, spectral, or multiscale distances. While most subgraph counting kernels are local [18], most random walk kernels are spectral [13]. Multiscale Laplacian [17], Weisfeiler Lehman kernel [19], and propagation kernel [20] are among the multiscale kernels.

In this paper, we introduce a graph kernel based on comparison of global Eulerian traversals of the two graphs. To the best of our knowledge, our formalism is the first to capture global architectures of the two graphs as well as their local structures. Our kernel is based on the graph traversal edit distance introduced in this paper. We show that a lower bound for GTED can be computed in polynomial time using the linear programming relaxation of the problem. In practice, the linear program often yields an integer solution, in which case the computed lower bound is actually equal to GTED.

2 Problem Definition

Due to diversity of applications, input graphs can be obtained as molecular structure graphs, social network graphs, systems biology networks, or sequence assembly graphs such as de Bruijn graphs [21], A-Bruijn graphs [22], positional de Bruijn graphs [23], string graphs [24], or implicit string graphs [25] among numerous alternatives. Our graph traversal edit distance is inspired by those

applications and can potentially be adapted to any of those frameworks. However, we choose below a general, convenient representative definition for the problem. For the sake of brevity, we assume throughout this paper that the input graph has one strongly connected component.

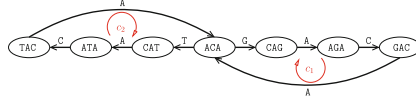


Fig. 1. Edge-labeled Eulerian graph. An edge-labeled Eulerian graph $A = (V, E, M, L, \{A, C, G, T\})$ obtained from the $k = 4$ de Bruijn graph $G = (V, E)$ for the circular sequence **ACAGACAT** [26]. Vertices, V , correspond to $(k - 1)$ -mers and edges correspond to k -mers. In this case, all the edges have multiplicity one, i.e. $M \equiv 1$. Edge labels, L , show the k^{th} nucleotide in the associated k -mers.

Definition 1 (Edge-labeled Eulerian Graph). Let Σ be a finite alphabet. We call a tuple $A = (V, E, M, L, \Sigma)$ an edge-labeled Eulerian graph, in which

- $G = (V, E)$ is a strongly connected directed graph,
- $M : E \rightarrow \mathbb{N}$ specifies the edge multiplicities,
- $L : E \rightarrow \Sigma$ specifies the edge labels,

iff G with the corresponding edge multiplicities, M , is Eulerian. That is, G contains a cycle (or path from a specified source to a sink) that traverses every edge $e \in E$ exactly $M(e)$ times. Throughout this paper, we mean M -compliant Eulerian by an Eulerian cycle (path) in A .

Figure 1 demonstrates an example edge-labeled Eulerian graph for the circular sequence **ACAGACAT** in the alphabet $\Sigma = \{A, C, G, T\}$. The sequence of edge labels over the Eulerian cycle formed by c_1 followed by c_2 yields the original sequence. The following definition makes a connection between Eulerian cycles and different sequences they spell.

Definition 2 (Eulerian Language). Let $A = (V, E, M, L, \Sigma)$ be an edge-labeled Eulerian graph. Define the word ω associated with an Eulerian cycle (path) $c = (e_0, \dots, e_n)$ in A to be the word

$$\omega(c) = L(e_0) \dots L(e_n) \in \Sigma^*. \quad (1)$$

The language of A is then defined to be

$$\mathcal{L}(A) = \{\omega(c) \mid c \text{ is an Eulerian cycle (path) in } A\} \subset \Sigma^*. \quad (2)$$

We now define graph traversal edit distance (GTED).

Problem 1 (Graph Traversal Edit Distance). Let A_1 and A_2 be two edge-labeled Eulerian graphs. We define the edit distance between A_1 and A_2 by

$$d(A_1, A_2) = \min_{\substack{\omega_1 \in \mathcal{L}(A_1) \\ \omega_2 \in \mathcal{L}(A_2)}} d(\omega_1, \omega_2), \quad (3)$$

in which $d(\omega_1, \omega_2)$ is the Levenshtein edit distance between two strings ω_1 and ω_2 . Throughout this paper, edit operations are single alphabet symbol insertion, deletion, and substitution, and the Levenshtein edit distance is the minimum number of such operations to transform ω_1 to ω_2 [27].

Note that $d(A_1, A_2)$ is the minimum of such edit distances over the words of possible Eulerian cycles (paths) in A_1 and A_2 . Note that GTED is almost a metric but not a metric since there are A_1, A_2 such that $d(A_1, A_2) = 0$ even though $A_1 \neq A_2$. For instance, let A_1 be an arbitrary Eulerian graph and A_2 be a cycle graph whose edge labels are the same as an arbitrary Eulerian cycle in A_1 . As a result, the graph traversal edit distance is different from the graph edit distance because the latter is a metric whereas the former is not.

3 Methods

3.1 Brute Force Computation of Graph Traversal Edit Distance

It is clear that there are algorithms, albeit with exponential running time, that enumerate all Eulerian cycles in a graph. Through brute force Needleman-Wunsch alignment of the words of every pair of Eulerian cycles in A_1 and A_2 , we can compute the edit distance right from the definition. De Bruijn, van Aardenne-Ehrenfest, Smith, and Tutte proved the de Bruijn-van Aardenne-Ehrenfest-Smith-Tutte (BEST) theorem [28, 29], which counts the number of different Eulerian cycles in A as

$$ec(A) = t_w(A) \prod_{v \in V} (\deg(v) - 1)!, \quad (4)$$

in which $t_w(A)$ is the number of arborescences directed towards the root at a fixed vertex w , and \deg is the indegree (or equally outdegree) considering multiplicities. The number of Eulerian cycles $ec(A)$ is exponentially large in general. Therefore, the naïve brute force algorithm is intractable.

3.2 Graph Traversal Edit Distance as a Constrained Shortest Path Problem

The conventional string alignment problem can be transformed into a shortest path problem in an alignment graph which is obtained by adding appropriate edges to the Cartesian product of the two string graphs. Figure 2 illustrates an example; further details can be found in a bioinformatics textbook such as [26]. Analogously, the graph traversal edit distance $d(A_1, A_2)$ can be written as the

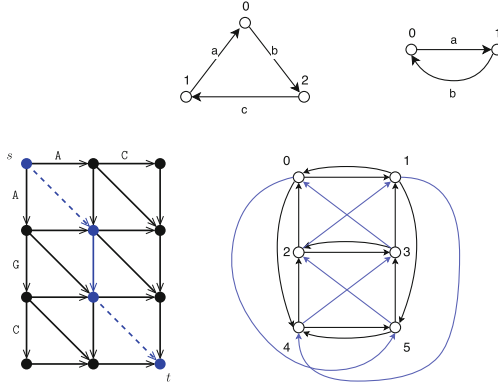


Fig. 2. Left: conventional alignment graph. The alignment graph for AC versus AGC. Those edges that correspond to matches are in dashed lines (cost of a match is often 0). Solid lines show substitutions and indels which usually have a positive cost. The edit distance is the shortest distance from s to t in this graph (shown in blue). **Right: example of an alignment graph.** The lower graph is an alignment graph for the two above graphs. Edges can have different costs, based on the edit operations for each pair of alphabets in the language. (blue edges correspond to math or mismatch and black edges correspond to insertion or deletions. (Color figure online)

length of the shortest cycle (or path from a designated source to a designated sink) in the alignment graph defined below, whose projection onto A_1 and A_2 is Eulerian. To state that fact in Lemma 1, we need

Definition 3 (Alignment Graph). Let $A_1 = (V_1, E_1, M_1, L_1, \Sigma_1)$ and $A_2 = (V_2, E_2, M_2, L_2, \Sigma_2)$ be two edge-labeled Eulerian graphs. Define the alignment graph between A_1 and A_2 to be $\mathcal{AG}(A_1, A_2) = (V_1 \times V_2, E)$, in which E is a collection of horizontal, vertical, and diagonal edges as follows:

- Vertical: $\forall e_1 = (u_1, v_1) \in E_1$ and $u_2 \in V_2 : e_1 \times u_2 = [(u_1, u_2), (v_1, u_2)] \in E$,
- Horizontal: $\forall u_1 \in V_1$ and $e_2 = (u_2, v_2) \in E_2 : u_1 \times e_2 = [(u_1, u_2), (u_1, v_2)] \in E$,
- Diagonal: $\forall e_1 = (u_1, v_1) \in E_1$ and $e_2 = (u_2, v_2) \in E_2 : [(u_1, u_2), (v_1, v_2)] \in E$.

There is a cost $\delta : E \rightarrow \mathbb{R}$ associated with each edge of \mathcal{AG} based on edit operation costs. Horizontal and vertical edges correspond to insertion or deletion and diagonal edges correspond to match or mismatch (substitution). A diagonal edge $[(u_1, u_2), (v_1, v_2)]$ is a match iff $L(u_1, v_1) = L(u_2, v_2)$ and a mismatch otherwise. We call A_i the i^{th} component graph. See Fig. 2 for an example.

The following Lemma states the fact that GTED is equivalent to a constrained shortest path problem in the alignment graph.

Lemma 1. For any two edge-labeled Eulerian graphs $A_1 = (V_1, E_1, M_1, L_1, \Sigma_1)$ and $A_2 = (V_2, E_2, M_2, L_2, \Sigma_2)$,

$$\begin{aligned}
 d(A_1, A_2) = & \underset{c}{\text{minimize}} \quad \delta(c) \\
 & \text{subject to} \quad c \text{ is a cycle (path) in } \mathcal{AG}(A_1, A_2), \\
 & \quad \pi_i(c) \text{ is an Eulerian cycle (path) in } A_i \text{ for } i = 1, 2,
 \end{aligned} \tag{5}$$

in which $\delta(c)$ is the total edge-cost (edit cost) of c , and π_i is the projection onto the i^{th} component graph.

Proof. For every pair (c_1, c_2) , in which c_i is an Eulerian cycle (path) in A_i , there are possibly multiple c 's with $\pi_i(c) = c_i$, whose minimum total edge-cost is $d(\omega(c_1), \omega(c_2))$. Therefore, the result of the minimization in (5) is not more than $d(A_1, A_2)$, i.e. the right hand side is less than or equal to $d(A_1, A_2)$. Conversely, every c that satisfies the constraints in (5) gives rise to an Eulerian pair $(c_1, c_2) = (\pi_1(c), \pi_2(c))$ and $\delta(c) \geq d(\omega(c_1), \omega(c_2)) \geq d(A_1, A_2)$, i.e. the right hand side is greater than or equal to $d(A_1, A_2)$.

3.3 Lower Bound via Linear Programming Relaxation

Lemma 1 easily transforms our problem into an integer linear program (ILP) as the projection operator π_i is linear and imposing path connectivity/cycle is also linear. More precisely, consider two edge-labeled Eulerian graphs A_1 and A_2 with the alignment graph $\mathcal{AG}(A_1, A_2) = (V_1 \times V_2, E)$, and let ∂ be the boundary operator, $\partial(e) = v - u$ for an edge $e = (u, v)$, which is defined in detail below. Our algorithm consists in solving the linear programming (LP) relaxation of that ILP,

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^{|E|}}{\text{minimize}} \quad \sum_{e \in E} x_e \delta(e) \\
 & \text{subject to} \quad \sum_{e \in E} x_e \partial(e) = 0 \quad (\text{or sink} - \text{source}), \\
 & \quad \forall e \in E, \quad x_e \geq 0, \\
 & \quad \text{for } i = 1, 2, \forall f \in E_i, \quad \sum_{e \in E} x_e I_i(e, f) = M_i(f),
 \end{aligned} \tag{6}$$

in which indicator function $I_1(e, f) = 1$ iff $e = f \times v_2$ or $e = [(u_1, u_2), (v_1, v_2)]$ with $f = (u_1, v_1)$; otherwise, $I_1(e, f) = 0$. Similarly, $I_2(e, f) = 1$ iff $e = v_1 \times f$ or $e = [(u_1, u_2), (v_1, v_2)]$ with $f = (u_2, v_2)$; otherwise, $I_2(e, f) = 0$. The linear program above is not guaranteed to give an integer solution; however, we have observed integer solutions in many scenarios. Nevertheless, the solution of (6) is a lower bound for GTED. Theoretically, both the lower bound and the exact GTED take polynomial time. However, the lower bound has a simpler linear program and is easier to implement, debug, back trace, and work with.

3.4 Algorithm for Graph Traversal Edit Distance

The following theorem is the main result of this paper which bridges the gap between GTED and another linear programming formulation which we will show is guaranteed to have an exact integer solution. Hence, GTED has a polynomial time algorithm explained as a linear program; Corollary 1 states that fact below.

Theorem 1 (GTED). *Consider two edge-labeled Eulerian graphs $A_i = (V_i, E_i, M_i, L_i, \Sigma_i)$ with $G_i = (V_i, E_i)$ for $i = 1, 2$. Let T be the collection of two-simplices in the triangulated $G_1 \times G_2$ with one-faces in $\mathcal{AG}(A_1, A_2)$. In that case,*

$$\begin{aligned} d(A_1, A_2) = \underset{x \in \mathbb{R}^{|E|}, y \in \mathbb{R}^{|T|}}{\text{minimize}} \quad & \sum_{e \in E} x_e \delta(e) \\ \text{subject to} \quad & x = x^{\text{init}} + [\partial] y, \\ & \forall e \in E, \quad x_e \geq 0, \end{aligned} \tag{7}$$

in which $[\partial]_{|E| \times |T|}$ is the matrix of the two-dimensional boundary operator in the corresponding homology and

$$x_e^{\text{init}} = \begin{cases} M_1(f) & \text{if } e = f \times s_2 \\ M_2(f) & \text{if } e = s_1 \times f \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

for arbitrary fixed $s_i \in V_i$ (source/sink in the case of path).

Proof. It is sufficient to show two things:

1. GTED is equal to the solution of the integer linear program (ILP) version of the linear program in (7),
2. the linear program in (7) always yields an integer solution.

Using Lemma 1, we need to show that (5) and (7) are equivalent for the first one. By construction, x^{init} corresponds to an Eulerian cycle (path) in A_1 followed by one in A_2 , which specifies a cycle (path) in $\mathcal{AG}(A_1, A_2)$ whose projection onto A_i is Eulerian. It is sufficient to note that every cycle (path) whose projection onto A_i is Eulerian is homologous to x^{init} . To see that, let c be a cycle (path) whose projection onto A_i is Eulerian. First note that diagonal edges in c are homologous to the horizontal edge followed by the vertical edge in the corresponding cell. Hence, diagonal edges can be replaced by the horizontal followed by the vertical edge using the boundary operator $[\partial]$. Hence without loss of generality, we assume c contains only horizontal and vertical edges.

If edges in c are $h_1, h_2, \dots, h_m, k_1, k_2, \dots, k_n$ such that $h_i = e_i \times s_2$ and $k_i = s_1 \times f_i$ for $e_i \in E_1$ and $f_i \in E_2$, then we are done. We know that such c has exactly the same representation as x^{init} . If edges in c are $h_1, h_2, \dots, h_m, k_1, k_2, \dots, k_n$ such that $h_i = e_i \times v_2$ and $k_i = v_1 \times f_i$ for $e_i \in E_1$ and $f_i \in E_2$ and possibly $v_1 \neq s_1$ or $v_2 \neq s_2$, then we can rotate the cycle through adding and subtracting a perpendicular translation edge and apply the boundary replacement operation to obtain a homologous cycle (path) of the form $h_1, h_2, \dots, h_m, k_1, k_2, \dots, k_n$

such that $h_i = e_i \times s_2$ and $k_i = s_1 \times f_i$ for $e_i \in E_1$ and $f_i \in E_2$. Starting with an arbitrary c , we show how to obtain a homologous cycle (path) in the form of $h_1, h_2, \dots, h_m, k_1, k_2, \dots, k_n$ such that $h_i = e_i \times v_2$ and $k_i = v_1 \times f_i$ for $e_i \in E_1$ and $f_i \in E_2$ through basic boundary replacement operations. Essentially, we show that we can swap vertical and horizontal edges along c until we end up with all horizontal edges grouped right up front followed by all vertical edges grouped at the end. Suppose c contains k, h as a subpath for $h = e \times v_2$ and $k = u_1 \times f$ and $e = (u_1, v_1) \in E_1$ and $f = (u_2, v_2) \in E_2$. The subpath k, h is homologous to h', k' in which $h' = e \times u_2$ and $k' = v_1 \times f$ since the four edges $k, h, -k', -h'$ form the boundary of a square. Hence, we can replace k, h with h', k' in c to obtain a homologous cycle (path) c' . Performing a number of such vertical-horizontal swaps will yield the result. The second is going to be shown in the following sections.

Corollary 1 (GTED complexity). *The graph traversal edit distance is in P and can be solved in polynomial time from the linear program in (7) that is guaranteed to give the integer solution.*

3.5 Total Unimodularity

Using a recent result of Dey et al. [30], we show that (7) is guaranteed to yield an integer solution. The main reason is that the boundary operator matrix $[\partial]$ is totally unimodular, i.e. all its square submatrices have a determinant in $\{0, \pm 1\}$. Therefore, all vertices of the constraint polytope in (7) have integer coordinates; hence, the solution is integer.

Why is $[\partial]$ totally unimodular? According to [30, Theorem 5.13], $[\partial]$ is totally unimodular iff the simplicial complex $G_1 \times G_2$ has no Möbius subcomplex of dimension 2. For the sake of completeness, we include the definition of a Möbius complex below.

Definition 4 ([30, Definition 5.9]). *A two-dimensional cycle complex is a sequence $\sigma_0 \cdots \sigma_{k-1}$ of two-simplices such that σ_i and σ_j have a common face iff $j = (i + 1) \bmod k$ and that the common face is a one-simplex. It is called a two-dimensional cylinder complex if orientable and a two-dimensional Möbius complex if nonorientable.*

Lemma 2. *A triangulated graph product space $G_1 \times G_2$ does not contain a Möbius subcomplex, for directed graphs G_i with unidirectional edges.*

Proof. It is enough to observe that in $G_1 \times G_2$, the orientation in one coordinate cannot flip. For brevity of presentation, we ignore triangulation for a moment and consider the rectangular cells. To the contrary, assume $G_1 \times G_2$ contains a Möbius subcomplex $\sigma_0 \cdots \sigma_{k-1}$ in which every σ_i is a rectangle $e_i \times f_i$, for $e_i \in E_1$ and $f_i \in E_2$. Since every σ_i and σ_{i+1} have a common edge and G_1, G_2 are directed graphs with unidirectional edges, either $e_{i+1} = e_i$ or $f_{i+1} = f_i$ but not both. In particular, $e_0 = e_{k-1}$ or $f_0 = f_{k-1}$. That is a contradiction because $\sigma_0 \cdots \sigma_{k-1}$ is then a cylinder subcomplex (orientable) and not a Möbius subcomplex.

Lemma 2 together with [30, Theorem 5.13] assert that $[\partial]$ is totally unimodular. Therefore, (7) always has an integer solution, hence the main result in Theorem 1.

Lack of Möbius subcomplexes in the product space of graphs, which are Möbius-free spaces, can also be seen from the fact that the homology groups of graph product spaces are torsion-free. The following section summarizes that characterization.

3.6 Homology Theory of Alignment Graph

An alignment graph $\mathcal{AG}(A_1, A_2)$ is essentially a topological product space with additional triangulating diagonal edges corresponding to matches and mismatches. In other words, $\mathcal{AG}(A_1, A_2)$ can be regarded as a triangulation of the two-dimensional CW complex $G_1 \times G_2$ (by horizontal, vertical, and diagonal edges). Note that $G_1 \times G_2$ has zero-dimensional vertices (v_1, v_2) , one-dimensional edges $e_1 \times v_2$ and $v_1 \times e_2$, and two-dimensional squares $e_1 \times e_2$ for $v_i \in V_i$ and $e_i \in E_i$. We characterize below the homology groups of $G_1 \times G_2$ using the Künneth's theorem. Note that G_i are obtained from edge-labeled graphs $A_i = (V_i, E_i, M_i, L_i, S_i)$.

Theorem 2 (Künneth [31]). *For graphs $G_i = (V_i, E_i)$, $i = 1, 2$,*

$$\begin{aligned}
 H_m(G_1 \times G_2, \mathbb{Z}) \cong & \bigoplus_{p+q=m} H_p(G_1, \mathbb{Z}) \otimes H_q(G_2, \mathbb{Z}) \\
 & \bigoplus_{r+s=m-1} \text{Tor}(H_r(G_1, \mathbb{Z}), H_s(G_2, \mathbb{Z})),
 \end{aligned}
 \tag{9}$$

in which H_m is the m^{th} homology group and Tor is the torsion functor [31].

Since $G_1 \times G_2$ is a two-dimensional CW complex, $H_m(G_1 \times G_2, \mathbb{Z}) \cong 0$ for $m > 2$. Clearly, $H_0(G_1 \times G_2, \mathbb{Z}) \cong \mathbb{Z}$ since $G_1 \times G_2$ is connected. According to the Künneth's theorem above and the fact that $\text{Tor}(\mathbb{Z}, \mathbb{Z}) \cong 0$ and $\mathbb{Z}^k \otimes \mathbb{Z} \cong \mathbb{Z} \otimes \mathbb{Z}^k \cong \mathbb{Z}^k$ [32],

$$\begin{aligned}
 H_1(G_1 \times G_2) \cong & [H_1(G_1) \otimes H_0(G_2)] \oplus [H_0(G_1) \otimes H_1(G_2)] \oplus \text{Tor}(H_0(G_1), H_0(G_2)) \\
 \cong & [\mathbb{Z}^{n_1} \otimes \mathbb{Z}] \oplus [\mathbb{Z} \otimes \mathbb{Z}^{n_2}] \oplus \text{Tor}(\mathbb{Z}, \mathbb{Z}) \cong \mathbb{Z}^{n_1} \oplus \mathbb{Z}^{n_2} \cong \mathbb{Z}^{n_1+n_2},
 \end{aligned}
 \tag{10}$$

in which $n_i = 1 + |E_i| - |V_i|$. Note that $H_1(G_1, G_2)$ is torsion-free.

Using the Künneth's theorem above and the fact that the tensor product of groups \otimes distributes over the direct sum \oplus , $H_2(G_i) \cong 0$, Tor of torsion-free groups is trivial, and $\mathbb{Z} \otimes \mathbb{Z} \cong \mathbb{Z}$, we obtain

$$\begin{aligned}
 H_2(G_1 \times G_2) \cong & [H_1(G_1) \otimes H_1(G_2)] \oplus \\
 & \text{Tor}(H_1(G_1), H_0(G_2)) \oplus \text{Tor}(H_0(G_1), H_1(G_2)) \\
 \cong & [\mathbb{Z}^{n_1} \otimes \mathbb{Z}^{n_2}] \oplus \text{Tor}(\mathbb{Z}^{n_1}, \mathbb{Z}) \oplus \text{Tor}(\mathbb{Z}, \mathbb{Z}^{n_2}) \\
 \cong & \bigoplus_{i=1}^{n_1} \bigoplus_{j=1}^{n_2} \mathbb{Z} \otimes \mathbb{Z} \cong \mathbb{Z}^{n_1 n_2}.
 \end{aligned}
 \tag{11}$$

Note that $H_2(G_1, G_2)$ is torsion-free.

4 Experiments

4.1 Using GTED to Make a Kernel

As mentioned earlier, since GTED is a measure of distance or dissimilarity between two graphs, we can use it to make a kernel of distance of pair of graphs in a dataset, and this can be used for classification problems. We implemented a C++ program that generates the linear program for the problem. First, it builds the alignment graph \mathcal{AG} for two given graphs $A_1 = (V_1, E_1)$ and $A_2 = (V_2, E_2)$ where V_i and E_i are vertices and edges of the i th graph. It begins with $|V_1| \times |V_2|$ vertices that are labeled as (v_1, v_2) for each $v_1 \in V_1$ and $v_2 \in V_2$. For each edge $(u_1, v_1) \in E_1$ and vertex $u_2 \in V_2$ we add the vertical edge $[(u_1, u_2), (v_1, u_2)]$ with a gap penalty δ_1 to our grid, \mathcal{AG} . We also add a horizontal edge $[(u_1, u_2), (u_1, v_2)]$ for each vertex $u_1 \in V_1$ and edge $(u_2, v_2) \in E_2$ with the same cost δ_1 . Then, for each pair of edges $(u_1, v_1) \in E_1$ and $(u_2, v_2) \in E_2$ we add a diagonal edge $[(u_1, u_2), (v_1, v_2)]$, with a mismatch penalty δ_2 if (u_1, v_1) has a different label from (u_2, v_2) , or a match bonus δ_3 if the labels are the same. The cost values are taken as arguments, with default values of $\delta_1 = \delta_2 = 1$ and $\delta_3 = 0$. This can be further extended to different penalties for insertion and deletion (i.e. different cost for horizontal and vertical edges).

The C++ program also creates a projection set for each edge in either of the input graphs. Each vertical edge $[(u_1, u_2), (v_1, u_2)]$ is added to the projection set of the edge $(u_1, v_1) \in E_1$, each horizontal edge $[(u_1, u_2), (u_1, v_2)]$ to the set of $(u_2, v_2) \in E_2$, and each diagonal edge $[(u_1, u_2), (v_1, v_2)]$ to projection sets of both $(u_1, v_1) \in E_1$ and $(u_2, v_2) \in E_2$.

Our program then extracts a linear programming problem from the alignment graph by assigning a variable x_i to the i th edge of \mathcal{AG} . The objective function minimizes weighted sum $\sum_{e \in E, \delta(e) > 0} x_e \delta(e)$. Then, the constraints will be generated. There are two different groups of constraints. The first group forces the vertices of the grid to have the same number of incoming edges and outgoing edges, forcing the output to be a cycle in the alignment graph. The second group forces the size of the projection set for each edge of the input graphs to be equal to its weight in that input graph, forcing the projection of the output to be Eulerian in both input graphs.

We used an academic license of Gurobi optimizer to solve the linear program. Since the variables are already supposed to be non-negative, it was not necessary to add inequalities to the LP for this purpose.

Data. We tested our graph kernel on four data sets. The Mutag data set consists of “aromatic and heteroaromatic nitro compounds tested for mutagenicity.” Nodes in the graphs represent the names of the atoms. The Enzymes dataset is a protein graph model of 600 enzymes from BRENDA database which contains 100 proteins each from 6 Enzyme Commission top level classes (Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases and Ligases). Protein structures are represented as nodes, and each node is connected to three closest proteins on the enzyme. The NCI1 dataset is derived from PubChem website

[pubchem.ncbi.nlm.nih.gov] which is related to screening of human tumor (Non-Small Cell Lung) cell line growth inhibition. Each chemical compound is represented by their corresponding molecular graph where nodes are various atoms (Carbon, Nitrogen, Oxygen etc.) and edges are the bonds between atoms (single, double etc.). The class labels on this dataset is either active or inactive based on the cancer assay. The PTC dataset is part of Predictive Toxicology Evaluation Challenge. This dataset is composed of graphs representing chemical structure and their outcomes of biological tests for the carcinogenicity in Male Rats (MR), Female Rats (FR), Male Mice (MM) and Female Mice (FM). The task is to classify whether a chemical is POS or NEG in MR, FR, MM and FM in terms of carcinogenicity.

Pre-processing and Post-processing. We use the Chinese Postman algorithm to make the input graphs Eulerian by adding the minimum amount of weights to the existing edges of the graphs. For directed graphs, we can use them directly in our algorithm, but for undirected graphs, we consider two edges in opposite directions for each undirected edge, and treat the two created opposite edges as separate variables in our linear programming problem.

Because our method requires edge labels, for those datasets such as Enzymes that have no edge labels, we use the concatenation of the source node label and the destination node label to make a label for every edge. To make the direction of the edge irrelevant, when we are comparing the two edge labels to see whether they match, we check both the equality of label of one to the label of the other or to the reverse label of the other edge which is obtained by reversing ordering of the source and destination nodes.

After computing the distance value between each pair of graphs, we have higher values for more distant (less similar) graphs. To prepare a normalized kernel to be used in other implemented classifiers like SVM, we have to map initial values such that for more similar graphs we obtain higher values (1 for identical pairs). To make this transformation, we have used two simple methods, and for each dataset we have used both of them and chose the one that gives us the best results during the cross validation on the training set. Then, this chosen method is used on the test set to get the final accuracy. The first method is to use $f(x) = \frac{1}{x+1}$ as the map function. The second method is to use the function $f(x) = 1 - \frac{x-min}{max-min}$ to map the distance values. Here, the *max* and *min* show the maximum and minimum distance values that we have among all possible pairs of graphs. Since we get 0 for identical graphs, the *min* is always 0. Hence, the map function can be simplified to $f(x) = 1 - \frac{x}{max}$. Both methods will give us 1 for similar graphs that have GTED values of 0, and numbers between 0 and 1 for more distant graphs. The more distant the pair of graphs are, the less the corresponding value in the kernel will be. Table 1 presents the overall running times for computing the kernel for each benchmark dataset.

Table 1. Running time for kernel computations for graph pairs, which were distributed into a cluster of 80 computers. Graph pairs = $\frac{n(n-1)}{2}$, where n is the number of graphs.

Dataset	#Graphs	#Pairs	Chinese postman (sec)	Kernel computation (min)
MUTAG	188	17,578	3	3
Enzymes	600	179,700	50	35
NCI1	4110	8,443,995	300	1760
PTC	414	85,491	47	17

Results. To evaluate whether this method works well at capturing the similarity and classifying the graphs, we used some benchmark datasets that are used to compare the graph kernels. We compare the kernels by evaluating the accuracy of an SVM classifier that uses them for classification. We used the same settings as in [17] so we can compare our results with previously computed results for other kernels. In this setting, we split the data randomly to two parts, 80% for training and 20% for testing. Then, we computed results for 20 different splitting using different random seeds. It can be seen from the table below that for the Mutag [33] and Enzymes [10] datasets, our kernel outperforms the other kernels. In the results table, we copied the values in [17] for other kernels.

Kernel/Dataset	Mutag [33]	Enzymes [10]	NCI1 [34]	PTC [35]
WL [19]	84.50(± 2.16)	53.75(± 1.37)	84.76 (± 0.32)	59.97(± 1.60)
WL-Edge [18]	82.94(± 2.33)	52.00(± 0.72)	84.65(± 0.25)	60.18(± 2.19)
SP [14]	85.50(± 2.50)	42.31(± 1.37)	73.61(± 0.36)	59.53(± 1.71)
Graphlet [18]	82.44(± 1.29)	30.95(± 0.73)	62.40(± 0.27)	55.88(± 0.31)
p -RW [12]	80.33(± 1.35)	28.17(± 0.76)	TIMED OUT	59.85(± 0.95)
MLG [17]	84.21(± 2.61)	57.92(± 5.39)	80.83(± 1.29)	63.62 (± 4.69)
GTED	90.12 (± 4.48)	59.66 (± 1.84)	65.83(± 1.14)	59.08(± 2.11)

Analysis. As shown in the table, our kernel achieves a higher accuracy on the Mutag and Enzymes datasets but gets average result on PTC and relatively weaker result on NCI1, as compared to other methods. Actually, none of the existing kernels can get the best results on all different kinds of data because each kernel captures only some features of the graphs. The Eulerian traversals of the graphs can be very informative for some specific applications, like Mutag. The aromatic and heteroaromatic chemical compounds in Mutag mostly consist of connected rings of atoms. These constituent rings can give us a good measure of proximity of two compounds. Since the language of Eulerian traversals includes the traversal of these rings in each compound, finding the minimum distance between the strings of the languages (which are built by the labels of the nodes that represent the name of atoms) for two different compounds can provide a

measure of the similar structures that they contain. That is why we get the best result for this dataset using our kernel.

Similarly, GTED outperforms the other kernels in the enzymes dataset. The enzymes in this dataset have certain shapes consisting of various protein structures (the nodes), and the combination of the individual structures and the nearby proteins gives us a good sense of the structure of the enzyme. In this case, Eulerian cycles usually give us a good approximation for the general spatial structure of the enzyme which leads to a good score.

The algorithm performed less well on the NCI1 and PTC data sets. We are uncertain of why this is, but it seems likely that the critical properties of the relevant chemicals are not captured by the Eulerian traversal.

4.2 Using GTED on Genomic Data

As mentioned earlier, the original goal of GTED was to find the best alignment of two genomes using only the assembly graphs, without having to create an assembled sequence first. The common alignment methods that compute the Levenshtein edit distance cannot take many factors into account, like having trans-locations in the genome, or the fact that assembly graphs could have multiple Eulerian cycles. Our method finds the best alignment among all possible alignments for all possible pairs of reference genomes that can be derived from the assembly graphs. As a result, it gives us a good measure to compute the distance (or similarity) between genomic sequences, and hence a way to cluster a group of samples. Therefore, to evaluate our method on genomic data, we chose genomes of Hepatitis B viruses in five different vertebrates; the virus in two of them (Heron and Tinamou) belong to Avihepadnavirus genus, and the ones in three of them (Horseshoe bat, Tent-making bat, and Woolly monkey) belong to Orthohepadnavirus genus.

Pre-processing and post-processing. First, for each pair of sequences we wished to compare, we generated a *colored de Bruijn graph*, a de Bruijn graph (assembly graph) that combines multiple samples in a single assembly graph with k -mers from different samples identified using different colored edges. We then extracted the graph for each specific color (genome). The linear programming problem for this experiment is produced almost like before; the difference here is that instead of using the second set of constraints to enforce that all edges of the input graphs are used exactly as many times as their multiplicities (an Eulerian cycle), we add the absolute value of the difference of the number of times that an edge is used in the alignment graph and its original weight in the corresponding input graph to the objective function of the LP. This way, we try to minimize this difference but allow some discrepancies. The extra flexibility seems necessary in this case, because the input graphs are large and contain numerous sources of error: sequencing errors, using cutoffs for edges, and crude estimates of the weights of the edges based on the coverage of sequences in the colored de Bruijn graph mean that the edge multiplicities are not completely accurate.

Results. The whole pre-processing step and generating the results took 4 h on 30 CPUs for each pair of viruses. Numbers in the table below are the computed distance of each of these pairs of graphs. As represented in the table, it can be seen that the intra-genus distances are lower than inter-genus distances. We believe, based on these numbers, a good estimate of the similarity of the genomes can be made, both for genomes in the same genus and the ones with various genus.

	Heron	Tinamou	Horseshoe bat	Tent-making bat	W. monkey
Heron	-	1016	1691	1639	1659
Tinamou	1016	-	1699	1638	1640
Horseshoe bat	1691	1699	-	1347	1296
Tent-making bat	1639	1638	1347	-	1429
Woolly monkey	1659	1640	1296	1429	-

5 Conclusion

In this paper we have introduced GTED, a new method for comparing networks based on a traversal of their edge labels. We have shown that GTED admits a polynomial time algorithm using a linear program. This linear program is guaranteed to have an integer solution due to the fact that the boundary operator function is totally unimodular, giving us an exact solution for the minimum possible edit distance.

The GTED problem was originally designed to be a formalization of the differential genome assembly problem, comparing DNA assembly graphs by considering all their possible assembled strings. It performs well at that task, successfully differentiating different genera of the Hepatitis B virus. We tested GTED on viral genomes since GTED is a global alignment scheme that is not immediately scalable to full-size large genomes, like all other global alignment schemes such as Needleman-Wunsch. However, GTED can form the mathematical basis for scalable heuristic comparison of full-size large genomes in the future. GTED can also be used as a general graph kernel on other types of networks, performing particularly well on graphs whose Eulerian traversals provide a good insight into their important structural features.

GTED is a new way of measuring the similarity between networks. It has many applications in differential genome assembly, but it also performs well in domains beyond assembly graphs. GTED has the potential to be a valuable tool in the study of biological networks.

References

1. Li, Y., et al.: Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome *de novo* assembly. *Nat. Biotechnol.* **29**, 723–730 (2011)
2. Movahedi, N.S., Forouzmand, E., Chitsaz, H.: De novo co-assembly of bacterial genomes from multiple single cells. In: *IEEE Conference on Bioinformatics and Biomedicine*, pp. 561–565 (2012)
3. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., McVean, G.: De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.* **44**, 226–232 (2012)
4. Taghavi, Z., Movahedi, N.S., Draghici, S., Chitsaz, H.: Distilled single-cell genome sequencing and de novo assembly for sparse microbial communities. *Bioinformatics* **29**(19), 2395–2401 (2013)
5. Movahedi, N.S., Embree, M., Nagarajan, H., Zengler, K., Chitsaz, H.: Efficient synergistic single-cell genome assembly. *Front. Bioeng. Biotechnol.* **4**, 42 (2016)
6. Hormozdiari, F., Hajirasouliha, I., McPherson, A., Eichler, E., Sahinalp, S.C.: Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res.* **21**, 2203–2212 (2011)
7. Mak, C.: Multigenome analysis of variation (research highlights). *Nat. Biotechnol.* **29**, 330 (2011)
8. Jones, S.: True colors of genome variation (research highlights). *Nat. Biotechnol.* **30**, 158 (2012)
9. Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: mining graph data. *Mach. Learn.* **50**(3), 321–354 (2003)
10. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.-P.: Protein function prediction via graph kernels. *Bioinformatics* **21**(1), 47–56 (2005)
11. Kubinyi, H.: Drug research: myths, hype and reality. *Nat. Rev. Drug Discov.* **2**(8), 665–668 (2003)
12. Gartner, T.: Exponential and geometric kernels for graphs. In: *NIPS 2002 Workshop on Unreal Data, Principles of Modeling Nonvectorial Data* (2002)
13. Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph kernels. *J. Mach. Learn. Res.* **11**, 1201–1242 (2010)
14. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In *Fifth IEEE International Conference on Data Mining (ICDM 2005)*, p. 8, November 2005
15. Feragen, A., Kasenburg, N., Petersen, J., de Bruijne, M., Borgwardt, K.: Scalable kernels for graphs with continuous attributes. In: *Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems*, vol. 26, pp. 216–224. Curran Associates Inc. (2013)
16. Kondor, R., Borgwardt, K.M.: The skew spectrum of graphs. In: *Proceedings of the 25th International Conference on Machine Learning, ICML 2008*, pp. 496–503. ACM, New York (2008)
17. Kondor, R., Pan, H.: The multiscale laplacian graph kernel. In: *Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems*, vol. 29, pp. 2990–2998. Curran Associates Inc. (2016)

18. Shervashidze, N., Vishwanathan, S.V.N., Petri, T., Mehlhorn, K., Borgwardt, K.: Efficient graphlet kernels for large graph comparison. In: van Dyk, D., Welling, M. (eds.) *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009*, vol. 5, pp. 488–495 (2009). PMLR
19. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.* **12**, 2539–2561 (2011)
20. Neumann, M., Garnett, R., Bauckhage, C., Kersting, K.: Propagation kernels: efficient graph kernels from propagated information. *Mach. Learn.* **102**(2), 209–245 (2016)
21. Pevzner, P.A., Tang, H., Waterman, M.S.: An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. U.S.A.* **98**, 9748–9753 (2001)
22. Pevzner, P.A., Tang, H., Tesler, G.: De novo repeat classification and fragment assembly. *Genome Res.* **14**(9), 1786–1796 (2004)
23. Ronen, R., Boucher, C., Chitsaz, H., Pevzner, P.: SEQuel: improving the accuracy of genome assemblies. *Bioinformatics* **28**(12), i188–i196 (2012). Also ISMB proceedings
24. Myers, E.W.: Toward simplifying and accurately formulating fragment assembly. *J. Comput. Biol.* **2**, 275–290 (1995)
25. Simpson, J.T., Durbin, R.: Efficient construction of an assembly string graph using the FM-index. *Bioinformatics* **26**, 367–373 (2010)
26. Jones, N.C., Pevzner, P.: *An Introduction to Bioinformatics Algorithms*. MIT press, Cambridge (2004)
27. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady* **10**(8), 707–710 (1966). Original. *Doklady Akademii Nauk SSSR* **163**(4), 845–848 (1965)
28. Tutte, W.T., Smith, C.A.B.: On unicursal paths in a network of degree 4. *Am. Math. Mon.* **48**(4), 233–237 (1941)
29. van Aardenne-Ehrenfest, T., de Bruijn, N.G.: Circuits and trees in oriented linear graphs. In: Gessel, I., Rota, G.-C. (eds.) *Classic Papers in Combinatorics, Modern Birkhäuser Classics*, pp. 149–163. Birkhäuser, Boston (1987)
30. Dey, T., Hirani, A., Krishnamoorthy, B.: Optimal homologous cycles, total unimodularity, and linear programming. *SIAM J. Comput.* **40**(4), 1026–1044 (2011)
31. Vick, J.W.: *Homology Theory: An Introduction to Algebraic Topology*, vol. 145. Springer, New York (1994). <https://doi.org/10.1007/978-1-4612-0881-5>
32. Massey, W.: *A Basic Course in Algebraic Topology*, vol. 127. Springer, New York (1991)
33. Debnath, A.K., de Compadre, R.L.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.* **34**(2), 786–797 (1991)
34. Wale, N., Watson, I.A., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.* **14**(3), 347–375 (2008)
35. Toivonen, H., Srinivasan, A., King, R.D., Kramer, S., Helma, C.: Statistical evaluation of the predictive toxicology challenge 2000–2001. *Bioinformatics* **19**(10), 1183–1193 (2003)



Statistical Inference of Peroxisome Dynamics

Cyril Galitzine¹(✉), Pierre M. Jean Beltran², Ileana M. Cristea²,
and Olga Vitek¹

¹ College of Science, College of Computer and Information Science,
Northeastern University, Boston 02115, USA
c.galitzine@northeastern.edu

² Lewis Thomas Laboratory, Department of Molecular Biology,
Princeton University, Washington Road, Princeton, NJ 08544, USA

Abstract. The regulation of organelle abundance sustains critical biological processes, such as metabolism and energy production. Biochemical models mathematically express these temporal changes in terms of reactions, and their rates. The rate parameters are critical components of the models, and must be experimentally inferred. However, the existing methods for rate inference are limited, and not directly applicable to organelle dynamics.

This manuscript introduces a novel approach that integrates modeling, inference and experimentation, and incorporates biological replicates, to accurately infer the rates. The approach relies on a biochemical model in form of a stochastic differential equation, and on a parallel implementation of inference with particle filter. It also relies on a novel microscopy workflow that monitors organelles over long periods of time in cell culture. Evaluations on simulated datasets demonstrated the advantages of this approach in terms of increased accuracy and shortened computation time. An application to imaging of peroxisomes determined that fission, rather than *de novo* generation, is predominant in maintaining the organelle level under basal conditions. This biological insight serves as a starting point for a system view of organelle regulation in cells.

Keywords: Bayesian inference · Stochastic differential equation
Stochastic process · Particle filter · Organelles · Replicate
Peroxisomes

1 Introduction

Eukaryotic cells are organized into subcellular membrane-bound structures, such as the mitochondria, peroxisomes, and endosomes, known as *organelles* (Fig. 1). Dynamic control of organelle abundance is fundamental for cellular homeostasis, allowing cells to adapt to their environmental, metabolic, and energetic needs [1–4]. Genetic mutations that affect organelle dynamics are known to cause severe diseases in humans [5].

The understanding of organelle dynamics has been central in basic cell biology research. The processes inducing changes in organelle abundance are well-known. These include organelle production by fission and/or *de novo* biogenesis, as well as organelle destruction by fusion and/or degradation [6–8].

However, the integration of these individual processes into the overall control of organelle abundance remains unclear. Although genetic mutations and pharmacological interventions have provided insight into individual mechanisms, the uncovered pathways shared components, thus complicating the integration [1, 3, 4]. Development of a biochemical model of organelle dynamics is therefore a valuable approach for gaining biological insight into organelle regulation.

Biochemical models express temporal changes in organelle abundance in terms of basic mechanistic processes called *reactions*. Since organelle abundances are typically low (tens to hundreds), a stochastic biochemical model [9] is best suited to model their temporal evolution [10].

Stochastic biochemical models characterize reactions with *rate parameters*, which relate the speed of occurrence of the reaction to organelle counts. In complex biological systems, the rate parameters cannot be determined from first principles, and have to be inferred from experimental measurements collected over time. Here, we propose an integrated microscopy and computational method to infer the rates that regulate organelle abundance from time course organelle counts in cell culture, as we demonstrate for peroxisomes.

Peroxisomes are critical organelles required for cell detoxification and lipid metabolism [3]. Fluorescence microscopy allows us to simultaneously count peroxisomes from multiple cells in the course of time in a minimally invasive manner. However, technological limitations restrict the experiments to less than 100 time points per cell, which for inference purposes is considered *sparse*. The counts are furthermore contaminated by biological and technological variation [11].

Here we argue that, similarly to any other area of data-driven research, rate inference in sparse settings is improved by conducting experiments with multiple cell replicates. Although extending the biochemical models of organelle regulation to replicated experiments is straightforward in theory, it is challenging in practice. First, the replicates complicate modeling and inference of rate parameters, as expressing cellular heterogeneity dramatically increases the computational cost. Second, long-term imaging of organelles (for over 8 h) is required to observe consistent changes in counts across cells. This is difficult to do for a single cell, and even more so for multiple cells. To our knowledge, there are no reports of peroxisome imaging for this length of time. As a result, previous studies are limited, focused on simulated data [12] or on transcription [13]. They are not applicable to studies of organelle dynamics.

To overcome the limitations above, we describe an algorithm for inferring rate parameters in biochemical models from replicated experiments, and an imaging method that supports the inference by long-term monitoring of peroxisome counts in multiple live cells. This algorithm takes as input peroxisome counts, acquired from fluorescent microscopy images by a commercial software. We demonstrate that this approach provides new biological insight into the mechanisms of peroxisome regulation.

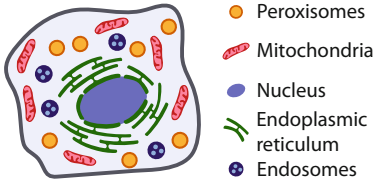


Fig. 1. Illustration of organelles in a eukaryotic cell.

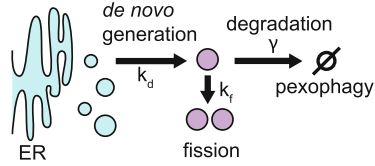


Fig. 2. The biochemical model that governs peroxisome count in a cell. Peroxisomes are created *de novo* at rate k_d or via fission at a rate k_f , and degraded at rate γ .

2 Background

2.1 Organelle Dynamics via Fluorescence Microscopy

Organelle dynamics is defined as the process that regulates organelle shape and numbers. Studies of organelle dynamics commonly use fluorescent probes and microscopy (abbreviated to fluorescence microscopy). The technology induces cells to produce a fluorescent protein fusion that is targeted to the organelle. Using fluorescent probes as markers, organelle structures are identified [2] and used to count organelle numbers [14].

Peroxisomes are particularly well suited for studies of dynamics, as their round punctate structure (Fig. 1) facilitates counting from microscopy images [3]. In addition, peroxisomes do not undergo fusion. The biochemical model that governs peroxisome counts in a cell is simplified to only three stochastic processes, each with its own rate parameter, as in Fig. 2. For example, an increase in peroxisome counts implies that the joint rate of processes that control biogenesis (i.e., *de novo* generation and fission) exceeded the rate of degradation. This sheds light into their involvement in cellular events that require changes in peroxisome numbers, such as during cell growth in human cells or in response to different nutrient conditions in yeast [10, 14].

For accurate rate inference, important data considerations include the availability of accurate counts, multiple replicates, and time lapse acquisition. However, live peroxisome imaging is commonly performed over small periods of time (a few minutes) [15], and high-throughput peroxisome imaging has been limited to the use of fixed cells [16]. Moreover, high magnification objectives (60X or higher) used to resolve peroxisomes ($0.5 - 1\mu$ m in size) [14] limit imaging to individual cells. This manuscript addresses these challenges by developing a dedicated experimental approach that allows imaging of 100 time points per cell and up to 20 replicate cells per experiment over a time period of over 8 h. This in turn enables the accurate inference of the rates.

2.2 Modeling and Inference of Organelle Dynamics

Modeling. Mukherji and O’Shea have proposed a stochastic model of organelle dynamics in yeast [10], which we review in this section in the case of peroxisomes. We denote by $x(t) \in \mathbb{N}$ the count of peroxisomes in a cell at time $t \in \mathbb{R}^+$. Given the joint effect of the three stochastic processes, the probability $p(x, t)$ that the count equals x at time t is governed by

$$\frac{dp(x, t)}{dt} = [k_d + k_f(x - 1)]p(x - 1, t) + [\gamma(x + 1)]p(x + 1, t) - [k_d + (k_f + \gamma)x]p(x, t) \quad (1)$$

where $p(x_0, 0) = 1$, and $p(x \neq x_0, 0) = 0$. The equation describes the Markov jump process [17], and is used in many areas of research, e.g. to describe a birth-death immigration process [17] in ecological systems [18]. The rate parameter k_d is in units of time^{-1} , while k_f and γ are in units of $\text{peroxisome}^{-1}\text{time}^{-1}$. In Eq. (1) k_d, k_f and γ are unknown and must be inferred from the data.

The data $\mathcal{D} = \bigcup_{t=1}^T (t_i, y_t)$ are time points $t_1 < t_2 < \dots < t_T$ and organelle counts y_1, \dots, y_T observed in a same cell. In presence of measurement error, the observed counts differ from the true (*hidden*) counts x_t governed by Eq. (1). The Normally distributed measurement error is frequently assumed $p(y_t | x_t) = \mathcal{N}(x_t, \sigma^2)$ [12, 19].

Inference. To infer the rate parameters in Eq. (1), traditional inference methods, such as those based on a particle filter [20], simulate many different trajectories from the equation via an exact simulation method, e.g. the Gillespie algorithm [21]. Unfortunately, these methods are computationally expensive. In this manuscript we propose to reformulate Eq. (1) in terms of an equivalent stochastic differential equation, which leads to Bayesian formulation and to inference with parallelization. It reduces the computational by a large fraction (~ 30), thus enabling rapid feedback for follow-up biological investigations.

2.3 Bayesian Rate Inference in Unreplicated Systems

Inference of stochastic biochemical systems is challenging because the likelihood $p(\mathcal{D} | \theta)$ is usually unavailable in closed form. Although frequentist modeling and inference has been proposed [22–24], it is less suited to experiments with sparse time-course measurements where the inferred rates are subject to relatively high uncertainty. Frequentist inference is therefore rarely used.

To our knowledge, the Bayesian formulation of Eq. (1) has never been considered. However, similar equations modeling other stochastic biochemical systems have received a great deal of attention in, e.g. [19, 25]. We briefly overview the approaches developed in these other contexts, as they form the basis of the proposed method for systems with multiple replicates.

Modeling. The Bayesian formulation of Eq. (1) requires the specification of a joint prior distribution of $\theta = (k_d, k_f, \gamma, \sigma)$, and the posterior

$$p(\theta | \mathcal{D}) \propto p(\mathcal{D} | \theta) p(\theta) \quad (2)$$

Assuming a memoryless process where the increments $x_{t+1} - x_t$ are statistically independent, and an error model where the measured count only depends on the hidden count, Eq. (2) becomes [19]

$$p(\theta | \mathcal{D}) \propto \int \prod_{t=1}^T p(y_t | x_t, \theta) p(x_t | x_{t-1}, \theta) p(\theta) dx_t \quad (3)$$

The integration is over all the possible hidden states at each time point.

Inference. Since the likelihood is unavailable in closed form, Bayesian inference is performed by numerically sampling from the posterior distribution. Most approaches are based on a Metropolis Hastings (MH) algorithm, but vary in methods that approximate the likelihood and update the parameters. For example, the exact stochastic process approximates the likelihood and an update scheme in [26]. A similar approach with the moment closure approximation is in [27]. However, these methods are inapplicable in presence of measurement error.

In presence of measurement error, the posterior distribution is most often sampled using a particle filter [20], which combines a Markov chain Monte Carlo (MCMC) sampler with a sequential Monte Carlo. It relies on the sequential propagation and reweighing of N computational particles $\mathbf{p}_{1 \leq i \leq N}$. Each particle has a weight $\mathbf{p}_i(w)$ and a value $\mathbf{p}_i(x)$ along the time points. The particle filter method propagates and reweighs the particles along the time course, such that the likelihood at time t in Eq. (3) is the product of particle weight sums over all time points.

Several variants of particle filter aim to improve its computational efficiency. For example, the Particle Marginal Metropolis Hastings (PMMH) [28, 29] simultaneously targets both the parameters and the hidden counts, i.e. $p(x, \theta | \mathcal{D})$. This manuscript takes an approach similar to PMMH. However, since we are not interested in inference for the hidden counts, we target $p(\theta | \mathcal{D})$.

Particle filter is computationally expensive, particularly when used for complex equations such as Eq. (1). As such, they are often parallelized and run on distributed memory systems [30] (although, to the best of our knowledge, never for stochastic biochemical models). Most implementations split the computational particles between multiple processes, and iteratively propagate and reweigh the particles locally within each process [31]. Particles (or other information) is exchanged between the processes to avoid infrequent or local weight normalization. Different such schemes have been proposed, e.g. distributed resampling with non-proportional allocation (DRNA) [32] or local selection (LS) [33].

2.4 Bayesian Rate Inference in Replicated Systems

To the best of our knowledge, replicated experiments have not been previously used to infer rate parameters of organelle dynamics. Here we briefly discuss related methods proposed in the context of other stochastic biochemical systems.

Modeling. In experiments with replicate time courses, the data are a collection of time points and organelle counts across $k = 1, \dots, K$ cells. In the notation of this manuscript, $\mathcal{D} = \bigcup_{k=1}^K \mathcal{D}^k$, where $\mathcal{D}^k = \bigcup_{t=1}^{T_k} (t_t^k, y_t^k)$. The time steps $t_{t+1}^k - t_t^k$ can vary between the cells.

Zechner *et al.* model transcriptional and post-transcriptional processes in heterogeneous cell populations, where rates vary between cells [13, 34, 35]. Assuming that the replicate cells are governed by the same rates (*homogeneous* rates) and are statistically independent, the posterior in Eq. (3) becomes [13]:

$$p(\theta | \mathcal{D}) \propto \prod_{k=1}^K \int \prod_{t=1}^{T_k} p(y_t^k | x_t^k, \theta) p(x_t^k | x_{t-1}^k, \theta) p(\theta) dx_t^k \quad (4)$$

Since no information about rate values is known *a priori*, $p(\theta)$ is a weakly informative prior (e.g. a Lognormal distribution). Unfortunately, the method cannot handle situations where numbers of data points or measurement time steps differ between the cells. Therefore, this approach is unsuitable for inference of rate parameters of peroxisome dynamics from microscopy data. The likelihood of Eq. (4) can also be extended [13, 36] to a situations where the rates vary between cells (i.e., are *heterogeneous*) and statistically independent. In such case, the variation of rates between cells (i.e. the *intrinsic* biological variation) is described in terms of distribution $p(\theta | \alpha)$ with hyperparameters α . Expressing the posterior of Eq. (4) in terms of α , we obtain:

$$p(\alpha | \mathcal{D}) \propto \int \prod_{k=1}^K \int \prod_{t=1}^{T_k} p(y_t^k | x_t^k, \theta) p(x_t^k | x_{t-1}^k, \theta) p(\theta | \alpha) p(\alpha) dx_t^k d\theta \quad (5)$$

The rates are often assumed to follow a gamma distribution [36] which ensures their positivity and can well approximate the Normal distribution.

Inference. Zechner *et al.* [13] aimed to reconstruct promoter activation and transcription. Therefore, they were interested in the distribution over hidden counts $(x_1^1, \dots, x_T^1, \dots, x_1^K, \dots, x_T^K)$. They jointly inferred the hidden counts and the rate parameters by sampling from $p(x, \theta | \mathcal{D})$. Since targeting this distribution via Metropolis Hastings was intractable, the authors introduced a recursive Bayesian procedure where (ignoring cell to cell variations for the rates) the posterior distribution at time t_t was computed from the posterior distribution at time t_{t-1}

$$p(x_{1:t}^1, \dots, x_{1:t}^K, \theta | y_{1:t}^1, \dots, y_{1:t}^K) \propto \left[\prod_{k=1}^K p(y_t^k | x_t^k, \theta) p(x_t^k | x_{t-1}^k, \theta) \right] p(x_{1:t-1}^1, \dots, x_{1:t-1}^K, \theta | y_{1:t-1}^1, \dots, y_{1:t-1}^K)$$

until all the T time points of the K cell replicates have been used.

In contrast, studies of peroxisome dynamics are not interested in the hidden counts, and only need to sample $p(\theta | \mathcal{D})$ or $p(\alpha | \mathcal{D})$. This reduced dimensionality allows us to directly sample from Eqs. (4) or (5), without resorting to the complications of Eq. (6).

3 Methods

3.1 Expressing the Biochemical Model of Peroxisome Dynamics as a Stochastic Differential Equation

We propose to reformulate the model in Eq. (1) in terms of an equivalent stochastic differential equation (SDE) [37]

$$dx(t) = [k_d + (k_f - \gamma)x(t)]dt + [k_d + (k_f + \gamma)x(t)]dW(t) \text{ with } x(0) = x_0 \quad (6)$$

where $W(t)$ is Brownian motion, and $x(t) \in \mathbb{R}$ is the continuous approximation of the discrete peroxisome count $x(t) \in \mathbb{N}$. Obtained by the diffusion approximation, this equation has the same solution as Eq. (1) [38], but is less expensive to solve.

We solve this equation with the Euler-Maruyama method [39]. The solution advances with time step $\Delta t = t_{t+1} - t_t$ following:

$$x_{t+1} = [k_d + (k_f - \gamma)x_t]\Delta t + [k_d + (k_f + \gamma)x_t]\sqrt{\Delta t}Z \text{ with } Z \sim \mathcal{N}(0, 1) \quad (7)$$

to obtain x_{t+1} from x_t . In the following, the numerical solution of the SDE from t_t to t_{t+1} is abbreviated $x_{t+1} \sim p_{t_t \rightarrow t_{t+1}}(x_{t+1} | x_t, \theta)$. Since the solution is not deterministic, solving the equation between two time step amounts to sampling from the transition density between the steps.

Our experience indicates limited rate variation between the cells. We therefore first assume that all replicate cells in an experiment are homogeneous, i.e. have the same peroxisome regulation rates. We further assume no prior information about the rates, and specify a flat, uninformative prior $p(\theta) = 1$. In a second time, we relax the homogeneous rate assumption by considering heterogeneous rates, this time assuming a flat prior for the hyperparameters $p(\alpha) = 1$.

In fluorescence microscopy a variety of experimental factors, e.g. the luminosity of the fluorescent tag or the topology of each cell, impact the measurement error. We express this with a Normal measurement error, i.e.:

$$p_\varepsilon(y_t^k | x_t^k, \theta) = \frac{1}{\sqrt{2\pi}\sigma^k} e^{-\frac{1}{2(\sigma^k)^2}(y_t^k - x_t^k)^2} \quad (8)$$

where the standard deviation σ^k depends on the cell replicate k , but is constant in time.

Considering both the SDE model and the measurement error, and marginalizing the hidden states, the posterior analogous to Eq. (4) becomes:

$$p(\theta | \mathcal{D}) \propto \prod_{k=1}^K \int \prod_{t=1}^{T_k} [p_\varepsilon(y_t^k | x_t^k, \theta) p_{t_t^k \rightarrow t_{t+1}^k}(x_t^k | x_{t-1}^k, \theta)] dx_t^k \quad (9)$$

while in the case of heterogeneous rates, the posterior analogous to Eq. (5) is:

$$p(\alpha | \mathcal{D}) \propto \int \prod_{k=1}^K \int \prod_{t=1}^{T_k} p(\theta | \alpha) [p_\varepsilon(y_t^k | x_t^k, \theta) p_{t_t^k \rightarrow t_{t+1}^k}(x_t^k | x_{t-1}^k, \theta)] dx_t^k d\theta \quad (10)$$

Algorithm 1. *Metropolis Hastings Sampler*

Inputs: data \mathcal{D}
 Params: # of MCMC samples S
 # of burn in samples S_b
 initial values θ^0
 random walk parameter σ_{MH}
 Functions: Algorithm 2
 Output: samples $\{\theta^{S_b}, \dots, \theta^{S-1}\}$

```

1: procedure MCMCs( $\mathcal{D}$ )
2:   for  $s$  in  $0 : S - 1$  do
3:     Process Proc0 does:
4:       ▶ Generate proposal parameter
5:        $\theta^* \sim \text{Lognormal}(\log \theta^s, \sigma_{\text{MH}}^2)$ 
6:     All processes of  $\mathcal{P}^k$  collectively do:
7:        $\theta^{*,k} \leftarrow \theta^*$ 
8:       ▶ Calculate replicate log-likelihood
9:        $\text{LogLik}_k(\theta^{*,k}) \leftarrow \text{PPF}(\theta^{*,k}, \mathcal{D}^k, \mathcal{P}^k)$ 
10:    Process Proc0 does:
11:    ▶ Sum all replicate log-likelihoods
12:     $\text{LogLik}(\theta^*) \leftarrow \sum_{k=1}^K \text{LogLik}_k(\theta^{*,k})$ 
13:    ▶ Calculate MH acceptance ratio
14:     $\text{LogA} \leftarrow \text{LogLik}(\theta^*) - \text{LogLik}(\theta^s)$ 
15:     $\text{LogA} \leftarrow \text{LogA} + \log \frac{\prod_i \theta_i^*}{\prod_i \theta_i^s}$ 
16:    ▶ Accept/reject proposal
17:     $r \leftarrow \min(0, \text{LogA})$ 
18:     $u \sim U(0, 1)$ 
19:    if  $\log u < r$  then
20:       $\theta^{s+1} \leftarrow \theta^*$ 
21:    else
22:       $\theta^{s+1} \leftarrow \theta^s$ 

```

3.2 Parallel Inference for Replicated Experiments with Homogeneous Rates

MCMC Sampling. The reformulation of the model in Eq. (1) in terms of a SDE in Eq. (6) reduces the computational cost of parameter estimation. Specifically, we propose to sample the posterior distribution $p(\theta | \mathcal{D}) = p(k_d, k_f, \gamma, \sigma^1, \dots, \sigma^K | \mathcal{D})$ in Eq. (9) with the Metropolis Hastings algorithm.

The algorithm requires us to calculate the log likelihood $\text{LogLik}_k = \log [p(\theta | \mathcal{D}_k)] = \log [p(\mathcal{D}_k | \theta)]$ for each cell replicate k , and the overall log likelihood $\text{LogLik} = \sum_{k=1}^K \text{LogLik}_k$. The advantage of the algorithm is its ability to carry out the inference in a distributed memory multicore environment, and in a parallel manner. While traditional implementations of particle filter

approximate each LogLik_k in a single core, here we propose to simultaneously calculate LogLik_k using multiple computing cores or CPUs (called *processes* in what follows). The parallelization along each replicate is fairly natural and straightforward for the calculation of the overall log likelihood. The parallel calculation of each replicate log likelihood with a particle filter is, however, more involved due to the need to exchange particle between processes. This will reduce the computation time of each MCMC step, and in turn drastically reduce the overall computation time.

The proposed sampling is a modification of a standard Metropolis Hastings algorithm, as detailed in Algorithm 1. Global operations involving all cell replicates, such as the generation/acceptance of MH samples (lines 5 and 15), or the sum of LogLik_k (line 12) are standard, and performed by the master process Proc^0 . A Lognormal proposal distribution (lines 5 and 15) enforces the positivity condition for θ . The magnitude of each rate step (line 5) is proportional to the value of the rate.

Parallel Particle Filter. The calculation of LogLik_k with parallel particle filter (Algorithm 2) is the core of the proposed algorithm. It is an instance of distributed resampling with non-proportional allocation (DRNA), with global reweighing at each step [32, 40, 41]. Unlike the existing algorithms, we distribute the particles of a LogLik_k between multiple processes, and allow each process to resample its own particles. To facilitate mixing, a fraction of particles are exchanged between a process and its neighbors. We describe this in more detail below.

The algorithm partitions all the available processes (except the master Proc^0) into K groups. Every group $\mathcal{P}^k = \{\text{Proc}_0^k, \text{Proc}_1^k, \dots, \text{Proc}_{N_{\text{proc},k}}^k\}$ is dedicated to calculating LogLik_k . Proc_0^k is the master process used for global group operations, while the rest $N_{\text{proc},k}$ processes are slave processes.

Each slave process Proc_j^k stores N particles of the filter related to cell replicate k , denoted by $\mathbf{p}_i^{j,k}$, $1 \leq i \leq N$. Each particle has a weight, which characterizes the plausibility of its representation of the hidden state. The particle values $\mathbf{p}_i^{j,k}(x)$ are initialized from a Poisson distribution centered around the observed organelle counts at t_1 , and the particle weights $\mathbf{p}_i^{j,k}(w)$ from a Uniform distribution (lines 5–6). At each observed time point t the particles are propagated to $t + 1$ according to the Euler scheme (line 11 and Eq. (7)). This is the most computationally expensive part, due to the large number of particles considered.

After the update, each particle is re-weighted following the Normal error model (line 13 and Eq. (8)). Finally, the algorithm sums all the particle weights into the quantity SW (line 14), and increments the LogLik_k of cell k (line 18).

Algorithm 2. *Parallel Particle Filter*

Inputs: parameters θ
 data \mathcal{D}^k and processes \mathcal{P}^k for cell k
 Params: # of particles per process N
 Output: LogLik_k of cell k

```

1: procedure PPF( $\theta, \mathcal{D}^k, \mathcal{P}^k$ )
2:   Each Proc $_j^k$   $1 \leq j < N_{\text{proc},k}$  does:
3:      $\blacktriangleright$  Initialize particle values and weights
4:     for  $i$  in 1 to  $N$  do
5:        $\mathbf{p}_i^{j,k}(x) \sim \text{Pois}(y_1^k)$ 
6:        $\mathbf{p}_i^{j,k}(w) \leftarrow \frac{1}{N \times N_{\text{proc},k}}$ 
       LogLik $_k \leftarrow 0$ 
7:   for  $t$  in 0 to  $T^k - 1$  do
8:     Each Proc $_j^k$   $1 \leq j < N_{\text{proc},k}$  does:
9:       for  $i$  in 1 to  $N$  do
10:         $\blacktriangleright$  Propagate particles, Eq. (7)
11:         $\mathbf{p}_i^{j,k}(x) \sim p_{t \rightarrow t+1}(\cdot | \mathbf{p}_i^{j,k}(x), \theta)$ 
12:         $\blacktriangleright$  Calculate particle weights, Eq. (8)
13:         $\mathbf{p}_i^{j,k}(w) \leftarrow p_\varepsilon(y_{t+1}^k | \mathbf{p}_i^{j,k}(x), \theta)$ 
14:        Send  $\sum_i \mathbf{p}_i^{j,k}(w)$  to Proc $_0^k$ 
15:   Process Proc $_0^k$  does:
16:      $\blacktriangleright$  Increment LogLik $_k$ 
17:     SW  $\leftarrow \sum_{j=1}^{N_{\text{proc},k}-1} \sum_i \mathbf{p}_i^{j,k}(w)$ 
18:     LogLik $_k \leftarrow \text{LogLik}_k + \log\left(\frac{\text{SW}}{N}\right)$ 
19:   Each Proc $_j^k$   $1 \leq j < N_{\text{proc},k}$  does:
20:      $\blacktriangleright$  Exchange particles between processes
21:      $\left\{ \mathbf{p}_i^{j,k} \right\}_{i=N/2+1}^N \leftrightarrow \left\{ \mathbf{p}_i^{\text{Right}^{j,k},k} \right\}_{i=1}^{N/2}$ 
22:      $\left\{ \mathbf{p}_i^{j,k} \right\}_{i=1}^{N/2} \leftrightarrow \left\{ \mathbf{p}_i^{\text{Left}^{j,k},k} \right\}_{i=N/2+1}^N$ 
23:      $\blacktriangleright$  Normalize weights for each process
24:     for  $i$  in 1 to  $N$  do
25:        $\mathbf{p}_i^{j,k}(w) \leftarrow \frac{\mathbf{p}_i^{j,k}(w)}{\sum_i \mathbf{p}_i^{j,k}(w)}$ 
26:      $\blacktriangleright$  Resample particles by weight
27:     Sample  $\mathbf{p}_i^{j,k} \sim \mathbf{p}_i^{j,k}(w)$   $N$  times
28:   return LogLik $_k$ 

```

To prevent the loss of accuracy, the slave processes exchange particles in a circular manner, as illustrated in Fig. 3 (lines 21–22). For each process j of cell replicate k , $N/2$ particles are sent to the process Proc_{j-1}^k to its left, while the remaining half are sent to the process Proc_{j+1}^k to its right. The first process Proc_1^k is viewed as the neighbor of the last process $\text{Proc}_{N_{proc,k}}^k$. This ring topology minimizes the communication between the processes, and maximizes the efficiency of parallelization. Finally, after within-process weight normalization (line 25), the particles are sampled according to their weights using stochastic universal sampling [42] (line 27). This ensures that only highly plausible particles are retained for the next time step.

Since the calculation of each replicate likelihood is independent of the others, replicates with different number of data points and time discretization are trivially handled. If one cell replicate is acquired in a longer time course than the rest, it receives more processes to minimize the idle time of the other replicates waiting for the calculation.

Model-Based Conclusions. The inferred distribution of the rates are obtained from the output samples $(\theta^{S_b}, \dots, \theta^{S_b})$ of Algorithm 1. Since the samples are highly correlated, they are thinned by a factor (determined from their autocorrelation spectrum) before estimating the posterior distributions.

The units of k_f and γ differ from the units of k_d , and the values of the rates are not comparable directly. On the other hand, the ratios $k_d : k_f \bar{N} : \gamma \bar{N}$ (where \bar{N} is the average number of peroxisomes per cell) are the relative prevalence of each reaction in numbers of reaction per unit time. Therefore, to facilitate the interpretation, we report the results in terms of k_d , $k_f \bar{N}$ and $\gamma \bar{N}$ in what follows.

3.3 Inference of Cell to Cell Rate Variations

The method presented in the previous section can readily be extended to account for cell to cell variations in the rates. We assume that k_d , k_f and γ each follow a Gamma distribution with its own shape and rate parameters: i.e. $k_d \sim \text{Gamma}(\alpha_{k_d}, \beta_{k_d})$, $k_f \sim \text{Gamma}(\alpha_{k_f}, \beta_{k_f})$ and $\gamma \sim \text{Gamma}(\alpha_\gamma, \beta_\gamma)$. Instead of directly sampling the shape and scale of the Gamma distributions, we sample their mean μ and standard deviation σ . This approach is equivalent (since e.g. for k_d , $\alpha_{k_d} = \mu_{k_d}^2 / \sigma_{k_d}^2$ and $\beta_{k_d} = \sigma_{k_d}^2 / \mu_{k_d}$) but it allows a better interpretation of the inferred parameters, and reduces sampling variation. We propose

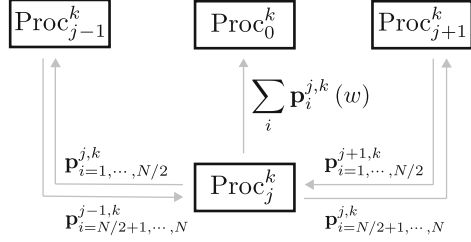


Fig. 3. Communication between Proc_j^k and its neighbors, at one time step, for cell replicate k . Proc_j^k sends the sum of particle weights $\sum_i \mathbf{p}_i^{j,k}(w)$ to the master node. Half of the particles are then exchanged between the neighboring slave processes.

to sample $\alpha = (\mu_{k_d}, \sigma_{k_d}, \mu_{k_f}, \sigma_{k_f}, \mu_\gamma, \sigma_\gamma, \sigma^1, \dots, \sigma^K)$ from the posterior distribution $p(\alpha | \mathcal{D})$ in Eq. (10) with the Metropolis Hastings algorithm detailed in the previous section. We approximate the integral over all θ parameters by using a single importance sample $\theta^k \sim p(\theta | \alpha)$ (which provides an unbiased estimator of the integral) so that:

$$p(\alpha | \mathcal{D}) \approx \prod_{k=1}^K \int \prod_{t=1}^{T_k} \left[p_\varepsilon(y_t^k | x_t^k, \theta^k) p_{t_t^k \rightarrow t_{t+1}^k}(x_t^k | x_{t-1}^k, \theta^k) \right] dx_t^k \quad (11)$$

We use the exact same particle filter detailed in Algorithm 2 while the difference in the MCMC sampler resides in that we sample α instead of θ and need to integrate the likelihood over θ . As such, in Algorithm 1, θ^* , θ^s , θ^{s+1} are replaced by α^* , α^s , α^{s+1} respectively while line 7 (the generation of rates from hyperparameters) becomes $\theta^{*,k} \sim p(\theta | \alpha)$.

3.4 Implementation

We implemented the procedure in C++ for speed, and parallelized it using the MPI-2.2 (Message Passing Interface) [43] libraries. The source code and documentation is available at github.com/cyrilgalitzine/Organelle.

3.5 Imaging and Counting Peroxisomes by Confocal Microscopy

First, peroxisomes in human liver cells (HepG2) were labeled by expression of the fluorescent protein, EGFP, tagged with the peroxisome targeting sequence, PTS1, as in [44]. Transfection conditions were optimized to avoid enlarged aberrant peroxisomes from overexpression, as well as reduce background cytosolic fluorescence while maintaining peroxisome-specific fluorescent signals. At 24 h following transfection, live cells were imaged with a 60X objective using a Nikon Ti-E confocal microscope. Z-stacks were acquired with 0.2 μm steps for 22 μm at 50 ms exposure per step to limit laser exposure to <10 s per cell. Image acquisition was automated for sequential imaging of individual cells (Fig. 4). Overall, this workflow maintained instrument use to a reasonable timeframe, and improved cell viability by avoiding continuous laser exposure. It maximized data collection at time intervals that allow detection of changes in peroxisome counts without oversampling. Using this instrumentation, we could image 20 cells with 6 min data point intervals for a total of 10 h.

To count peroxisomes, images were processed using the Nikon NIS-Elements AR v5.0. Image Z-stacks were deconvolved [45,46], and individual peroxisomes were detected semi-automatically using the 3D Spot Detection feature (Fig. 5). Organelle abundance was quantified as the number of objects detected per cell and per time point.

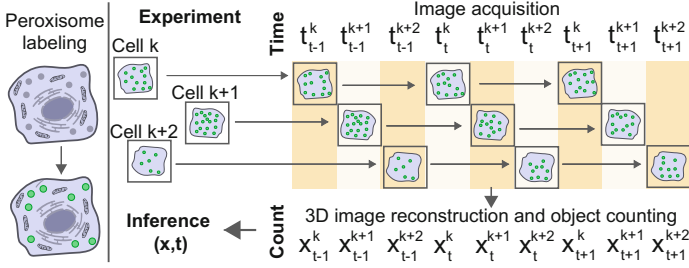


Fig. 4. Schematic representation of peroxisome imaging and counting in live human cells.

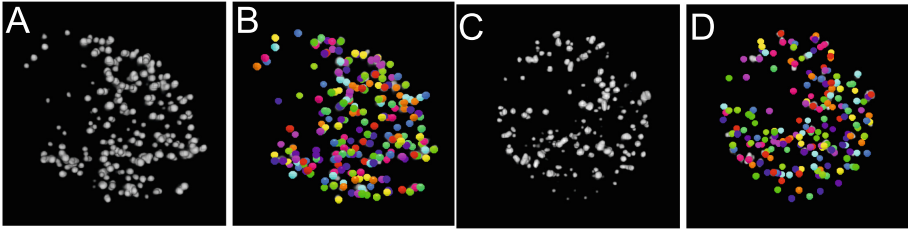


Fig. 5. A: peroxisome fluorescence signal from a cell. B: Output of the 3D spot detection algorithm for the same cell. Each colored sphere indicates a peroxisome. C, D: same as A, B, for another cell with fewer peroxisomes. (Color figure online)

4 Datasets

4.1 Experimental Datasets

We acquired a total of three experimental datasets, called Day 1, Day 2, and Day 3. The final datasets consisted of 13 replicate cells for Day 1, 10 replicate cells for Day 2, and 20 replicate cells for Day 3. The count results for two of these experiments are shown in Fig. 6. Between-cell and between-day variability was observed for both the average number of peroxisomes in a cell and the slope of the trace throughout the experiments. The number of cells per experiment and time points per cell varied as some cells moved out of focus or died before completion of the experiment. The cell heterogeneity and incomplete data were important considerations of the rate parameter inference.

4.2 Simulated Datasets

To evaluate the proposed approach in the case of homogeneous rates we simulated three additional datasets SIM A, SIM B and SIM C. The datasets were simulated with the Gillespie algorithm, with $K = 14$ and also with $K = 1$ cell replicates. Each cell replicate was initialized with a different count y_0^k (taken to be identical to the experimental counts of the first 14 cells on Day 3), but

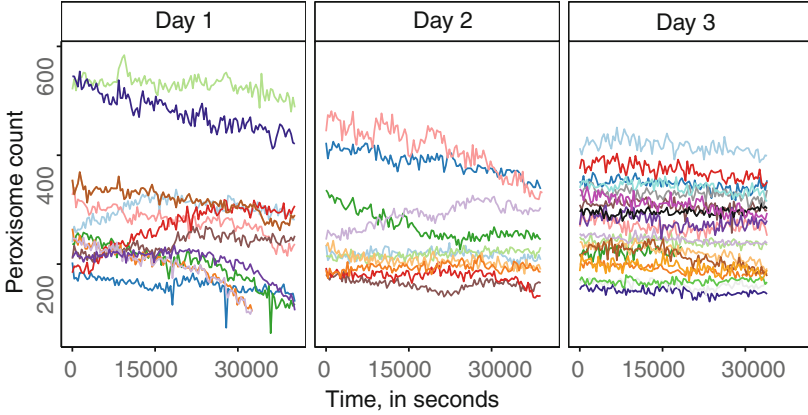


Fig. 6. Peroxisome counts in the time course experiments. Colors indicate cell replicates. (Color figure online)

simulated with the same duration $T^k = 33936$ s, frequency (336 s^{-1}) and standard deviation of the measurement error σ . At the inference stage, the standard deviation of the measurement error σ^k were inferred separately for each replicate.

The values of the rate parameters were inspired from those the experimental datasets, and reported in Table 1. In SIM A and SIM B the values of k_d were low, corresponding to a realistic situation where the *de novo process* is less prevalent than the fission or degradation, making it hard to detect. In SIM C k_d was relatively high, and the three reactions occurred relatively often. The values of k_f and γ were identical in SIM A, but differed slightly in SIM B and SIM C. The datasets were used to evaluate the ability of the proposed approach to estimate these different parameter configurations.

Table 1. Experimental design and parameter estimation in simulated datasets. True parameter values in each simulation are in bold. Table entries report parameter estimates, standard deviation of the posterior distribution (in parentheses), and the % error. k_d is expressed in second^{-1} , while k_f and γ in $\text{peroxisome}^{-1}\text{second}^{-1}$.

SIM	Dataset	$k_d \times 10^4$	$k_f \times 10^5$	$\gamma \times 10^5$	σ^1
A	True	7.75	4	4	6
	14 cells	29% 10 (7.65)	8% 4.32 (0.85)	12% 4.47 (0.85)	0.7% 5.95 (0.79)
	1 cell	1700% 140 (148)	4% 4.18 (3.05)	80% 7.17 (3.96)	-5% 5.68 (0.91)
B	True	2.75	3	4	7.5
	14 cells	224% 8.92 (5.50)	-9% 2.71 (0.93)	-1% 4.90 (0.91)	4% 7.8 (0.92)
	1 cell	758% 23.60 (26.5)	8% 3.01 (3.00)	13% 5.65 (3.20)	3% 7.70 (0.87)
C	True	50	1	5	6
	14 cells	-5% 47.2 (1.14)	30% 1.29 (0.63)	-4% 3.30 (0.58)%	-4% 5.77 (0.92)
	1 cell	8% 53.80 (51.6)	198% 2.98 (2.27)	73% 5.19 (2.34)	-10% 5.44 (0.71)

5 Results

5.1 Parallel Inference Shortened Computation Time

We inferred the rates in the experimental and the simulated datasets using 4800 particles and $\sigma_{\text{MH}} = 0.04$, resulting in a MH acceptance rate between 0.2 and 0.3. We set $N_{\text{proc},k} = 2$, and thinned the original 500,000 MCMC sampling iterations to every 500. Each iteration lasted on average 0.17s (wall clock time) for Day 1 (13 replicates on 40 CPUs), 0.14s for Day 2 (11 replicates on 34 CPUs) and 0.2s for Day 3 (20 replicates on 61 CPUs). The overall inference took around 1 day.

In contrast, the existing modeling and inference procedures required substantially longer computation time. For example, the use of the original stochastic model in Eq. (1) with Gillespie algorithm increased the time per iteration by about 30% with identical inference results. In the case of Day 1, this results in each iteration taking 0.2 seconds.

Similarly, the use of the SDE model in Eq. (6) with serial inference per cell increased the time per iteration by a factor of $K(N_{\text{proc},k} - 1)$ (the parallel overhead is negligible as compared to the particle movement). In the case of Day 1, representative iterations lasted 195s, estimating the overall inference time of 270 days. Therefore, the biological insights from multiple replicates were out of reach without the proposed parallel procedure.

5.2 The Approach Accurately Inferred the Rates

Figure 7 shows the posterior distributions, and Table 1 summarizes the properties of the inferred rates in the simulated datasets. Since the inferred σ^k were similar across replicates, the table only reports the value for the first replicate σ^1 . The proposed approach accurately inferred the rates in experiments with 14 cell replicates. SIM A and SIM B with low k_d challenged the estimation of this rate, as evidenced by its skewed and variable posterior distribution. A larger *de novo* rate in SIM C led to more accurate estimates. Table 1 shows that the inferred posterior distributions of $\bar{N}k_f$ and $\bar{N}\gamma$ had little variance. Their relatively large breadth in Fig. 7 was due to the multiplication by \bar{N} . We obtained identical inference results with the SDE model of Eq. (6) as with the Gillespie algorithm applied to Eq. (1) as shown in the case of SIM B (1 replicate) in Fig. 7. This, combined with the fact that simulation results were generated using Eq. (1), demonstrates that the SDE approximation reduced computational cost without compromising the accuracy of the results.

5.3 Replicate Cells Improved Inference of the Rates

Figure 7 compares the inferred posterior distributions with one versus 14 cell replicates in the simulated datasets, and Table 1 summarizes the results. Inference from unreplicated experiments had high uncertainty in all the experiments, and led to broader posterior distributions. In particular, rates associated with

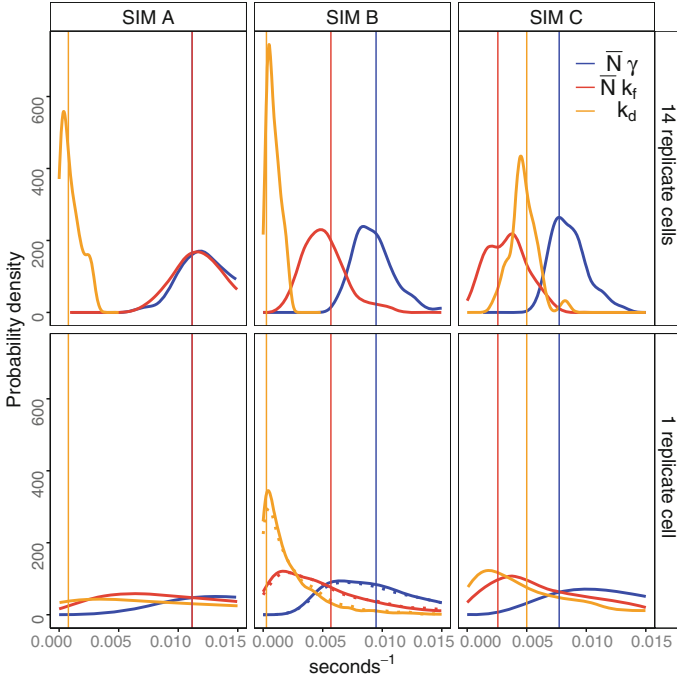


Fig. 7. Posterior distributions of the rates in the simulated datasets. To make the rates comparable, k_f and γ are multiplied by the average peroxisome count \bar{N} in each dataset. Vertical lines are the true parameter values. A dotted line denotes rate distributions obtained with Eq. (1) and the Gillespie algorithm instead of the SDE (solid line)

Table 2. Results for the experimental datasets with the homogeneous rate model. The reports parameter estimates, standard deviation of the posterior distribution (in parentheses). k_d is expressed in second^{-1} and k_f and γ in $\text{peroxisome}^{-1}\text{second}^{-1}$.

Dataset	$k_d \times 10^4$	$k_f \times 10^5$	$\gamma \times 10^5$
Day 1	2.51 (3.66)	10.0 (1.36)	10.5 (1.38)
Day 2	6.45 (5.60)	5.57 (1.12)	6.02 (1.11)
Day 3	1.05 (1.72)	1.19 (0.37)	1.43 (0.32)

rare events (such as *de novo* in SIM A and SIM B) could not be accurately estimated with only one replicate, and had an ~ 10 -fold error for the mean. The uncertainty diminished in experiments with 14 cell replicates, and the standard deviation of the posterior distributions of k_f and γ was reduced by a factor of 3 to 4. This result emphasized the importance of incorporating replicate cells into the rate inference procedure.

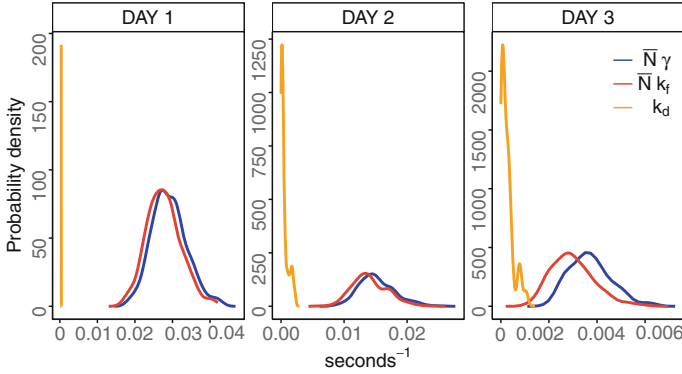


Fig. 8. As in Fig. 7, for the experimental datasets.

5.4 Inference from Replicate Cells Revealed Maintenance of Peroxisome Counts by Prominent Fission, Compared to Low *de novo* generation rates

Figure 8 compares the posterior distribution for the three experimental datasets, and Table 2 summarizes the results. We consistently observed similar values of rate parameters of peroxisomes degradation and fission. Moreover, we consistently observed a 5 to 100 times smaller value of the rate of *de novo* generation. The results indicate that *de novo* peroxisome generation in mammalian cells is a relatively rare event, occurring approximately 8 to 45 times per day.

5.5 Rates Varied Little Between Cells

We obtained in Sect. 5.4 fairly narrow inferred distribution for the rates which allowed us to make important biological conclusions. We would, however, like to distinguish how much of the variance of inferred rates is caused by possible rate heterogeneity between cells and how much is caused by statistical uncertainty (i.e. a too low number of replicates or data points). This is achieved by using the heterogeneous rate model which models rate cell to cell variations. Figure 9 shows the posterior hyperparameter distributions obtained with the heterogeneous rate model, and Table 3 summarizes the results. On average, for k_f and γ , the intrinsic rate standard deviation was about 5–10%, indicating relatively small intrinsic biological variations as compared to the rate values. The intrinsic variance of rates k_f and γ , i.e. σ_{k_f} and σ_γ , reported in Table 3 were less than 30% of the variance of the rate mean (in parentheses next to the mean value). This indicates that most of the variance obtained with the homogeneous rate model reported in Table 2 arised from statistical uncertainty (i.e. a too low number of replicate) instead of biological variation. In contrast, for days 1 and 3, σ_{k_d} was relatively large (about twice the rate mean standard deviation). This shows that, for some conditions, the uncertainty associated with cell to cell variation was more important than the statistical uncertainty in the *de novo* rate value.

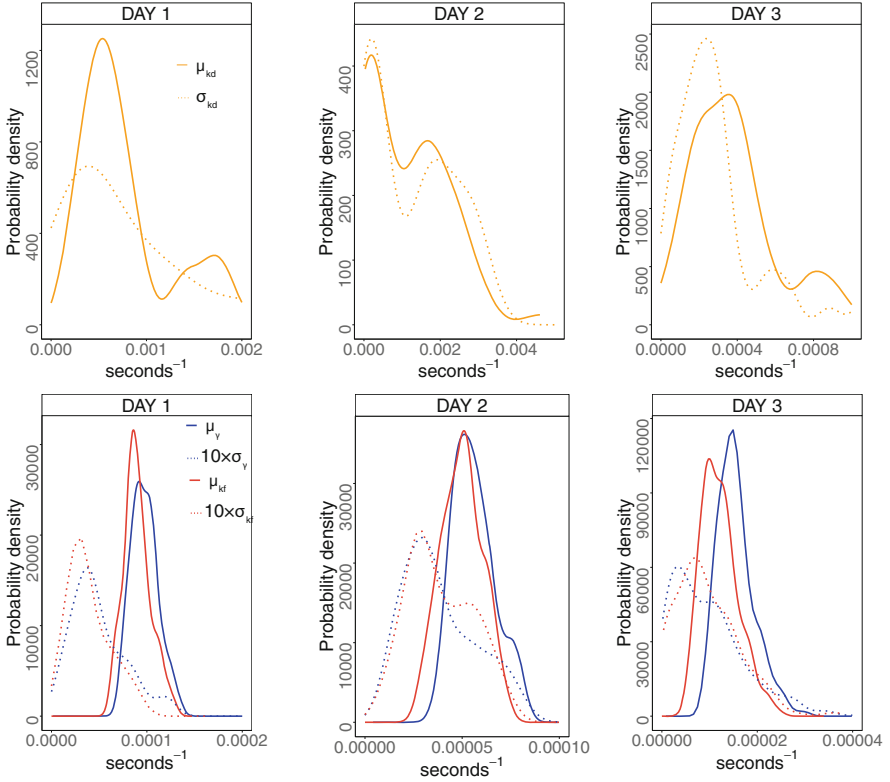


Fig. 9. As in Fig. 8, with the heterogeneous rate model. The rate standard deviations for k_f and γ (dotted lines) are multiplied by 10 to plot their distributions on the same plots as the rate means.

Table 3. Results for the experimental datasets for the heterogeneous rate model. The reports parameter estimates, standard deviation of the posterior distribution (in parentheses).

Dataset	$\mu_{k_d} \times 10^4$	$\sigma_{k_d} \times 10^4$	$\mu_{k_f} \times 10^5$	$\sigma_{k_f} \times 10^5$	$\mu_\gamma \times 10^5$	$\sigma_\gamma \times 10^5$
Day 1	7.6 (4.8)	8.9 (8.9)	8.99 (1.4)	0.39 (0.2)	9.75 (1.4)	0.49 (0.3)
Day 2	11.3 (10.5)	12.2 (10.9)	4.99 (1.03)	0.39 (0.16)	5.61 (1.06)	0.38 (0.18)
Day 3	4.6 (3.2)	3.5 (3.5)	1.21 (0.38)	0.12 (0.07)	1.55 (0.38)	0.096 (0.08)

6 Discussion

In contrast to other organelles, peroxisomes are constantly recycled in healthy cell populations, and degraded to remove old or damaged peroxisomes [47, 48]. Defining the predominant process of peroxisome production is a current topic of debate [49–52]. The stochastic model proposed by Mukherji and O’Shea was

tested in yeast cells [10]. The authors observed a switch from the predominance of *de novo* generation to fission, occurring under conditions that increase peroxisome numbers. While yeast cells only have 5–20 peroxisomes per cell, humans and other mammals need larger number of peroxisomes (100–500). It is therefore possible that mammals evolved to use fission as a primary mechanism for peroxisome proliferation [14, 47].

Here, we used experimental data to directly infer the rates governing peroxisome abundance. While the inferred rates for fission and degradation were similar, *de novo* generation was less frequent. The infrequent *de novo* generation is in line with previous studies estimating low numbers (30) of new peroxisomes per day [50]. The inference of peroxisome rates helps us reconcile previous conflicting evidence. It leads to a new model, where peroxisome population undergoes recycling via two opposing processes, fission and degradation, in addition to a basal *de novo* generation rate.

The results indicated that accuracy of rate inference depends on the value of the rates. In particular, rates associated with rare events, such as the *de novo* rate, are difficult to infer. This can be mitigated by imaging more cell replicates, or by extending the imaging time.

The inferred rates varied between instances of experiments repeated on multiple days. The variation in the rates across days could be explained as biological effects of the cell batch analyzed, such as confluency and age, which are known to affect the mechanisms of peroxisome biogenesis [53].

The cell to cell variation for the rates was limited in the case of the fission and degradation rates but more pronounced for the *de novo* rate which further compounded the uncertainty in its estimation.

The overall rate inference, and the assessment of the uncertainty, may be improved by analyzing the combined data from all the days. This will require extending the model to include inter-day variation, and accommodating the extra computational cost. Since the proposed SDE-based modeling is flexible, and since the inference algorithm supports parallelization, the proposed approach is in principle extendable to such situations. However, measurements on more days will be required to establish a model of inter-day variation.

This proposed inference procedure can also be extended to other organelles. Imaging tools for other organelles are available and widely used [44]. The modeling and inference procedure can include additional reactions, such as fusion.

Organelle dynamics are subject to alteration and regulation upon extracellular and intracellular cues. For example, peroxisomes increase in numbers when cells grow to undergo division. This occurs by either increasing the rate parameters of one of the production processes, or by decreasing the rate parameter of the degradation. The proposed approach can be used to assess this process of cell adaptation. Therefore, this work serves as a starting point for achieving a system view of the biophysical properties, used by the cell to regulate its organelle content.

Acknowledgements. This work was supported in part by a Burroughs Wellcome travel grant (to C.G.), a Dodds Fellowship (to P.M.J.B.), NIH grants GM114141, HL127640, Mallinckrodt Scholar Award (to I.M.C.), and Sy and Laurie Sternberg award (to O.V.).

References

1. Huotari, J., Helenius, A.: *EMBO J.* **30**, 3481 (2011)
2. Jean Beltran, P.M., Mathias, R.A., Cristea, I.M.: *Cell Syst.* **3**, 361 (2016)
3. Smith, J.J., Aitchison, J.D.: *Nat. Rev. Mol. Cell Biol.* **14**, 803 (2013)
4. Wai, T., Langer, T.: *Trends Endocrinol. Metab.* **27**, 105 (2016)
5. Steinberg, S.J., et al.: Peroxisome biogenesis disorders. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* **1763**(12), 1733–1748 (2006)
6. Denesvre, C., Malhotra, V.: *Curr. Opin. Cell Biol.* **8**, 519 (1996)
7. Lowe, M., Barr, F.A.: *Nat. Rev. Mol. Cell Biol.* **8**, 429 (2007)
8. Sica, V., et al.: *Mol. Cell* **59**, 522 (2015)
9. Wilkinson, D.J.: *Nat. Rev. Genet.* **10**, 122 (2009)
10. Mukherji, S., O’Shea, E.K.: *eLife* **3**, e02678 (2014)
11. Waters, J.C.: *J. Cell Biol.* **185**, 1135 (2009)
12. Amrein, M., Künsch, H.R.: *Stat. Comput.* **22**, 513 (2011)
13. Zechner, C., et al.: *Nat. Methods* **11**, 197 (2014)
14. Jauregui, M., Kim, P.K.: *Curr. Protoc. Cell Biol.* **62**, 21 (2014)
15. Costello, J.L., et al.: *J. Cell Biol.* **216**, 331 (2017)
16. Sexton, J.Z., et al.: *Int. J. High Throughput Screen* **2010**, 127 (2010)
17. Gillespie, D.T.: *Markov Processes: An Introduction for Physical Scientists*. Elsevier, Amsterdam (1991)
18. Hallam, T.G., Levin, S.A. (eds.): *Mathematical Ecology: An Introduction*. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-69888-0>
19. Wilkinson, D.J.: *Stochastic Modelling for Systems Biology*. CRC Press, Boca Raton (2011)
20. Doucet, A., Freitas, N.D., Gordon, N.: *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001). <https://doi.org/10.1007/978-1-4757-3437-9>
21. Gillespie, D.T.: *J. Comput. Phys.* **22**, 403 (1976)
22. Bretó, C., He, D., Ionides, E.L., King, A.A.: *Ann. Appl. Stat.* **3**, 319 (2009)
23. Ionides, E.L., Bretó, C., King, A.A.: *Proc. Natl. Acad. Sci.* **103**, 18438 (2006)
24. Wang, Y., et al.: *BMC Syst. Biol.* **4**, 1 (2010)
25. Fuchs, C.: *Inference for Diffusion Processes*. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-25969-2>
26. Boys, R.J., Wilkinson, D.J., Kirkwood, T.B.L.: *Stat. Comput.* **18**, 125 (2007)
27. Milner, P., Gillespie, C.S., Wilkinson, D.J.: *Stat. Comput.* **23**, 287 (2012)
28. Andrieu, C., Doucet, A., Holenstein, R.: *J. R. Stat. Soc. Ser. B* **72**, 269 (2010)
29. Golightly, A., Wilkinson, D.J.: *Interface Focus* **1**, 807 (2011)
30. Rosén, O., et al.: *IEEE CCA Proceedings*, p. 440. IEEE (2010)
31. Strid, I.: Technical report, Society for Computational Economics (2006)
32. Bolic, M., Djuric, P.M., Hong, S.: *IEEE Trans. Signal Process.* **53**, 2442 (2005)
33. Míguez, J., et al.: *EURASIP J. Adv. Signal Process.* **2004**, 303619 (2004)
34. Koepl, H., et al.: *Int. J. Robust Nonlin.* **22**, 1103 (2012)
35. Zechner, C., et al.: *Proc. Natl. Acad. Sci.* **109**, 8340 (2012)
36. Bronstein, L., Zechner, C., Koepl, H.: *Methods* **85**, 22 (2015)

37. Øksendal, B.: Stochastic Differential Equations. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-642-14394-6>
38. Gillespie, D.T.: J. Chem. Phys. **113**, 297 (2000)
39. Kloeden, P.E., Platen, E.: Numerical Solution of Stochastic Differential Equations (SDEs). Springer, Heidelberg (1992). <https://doi.org/10.1007/978-3-662-12616-5>
40. Míguez, J.: Signal Process. **87**, 3155 (2007)
41. Zenker, S.: J. Clin. Monit. Comput. **24**, 319 (2010)
42. Baker, J.E.: Proceedings of the 2nd International Conference on Genetic Algorithms, p. 14 (1987)
43. Pacheco, P.S.: Parallel Programming with MPI. Morgan Kaufmann, Burlington (1997)
44. Rizzo, M.A., et al.: Cold Spring Harbor Protoc. **4** (2009). <http://cshprotocols.cshlp.org/content/2009/12/pdb.top63.short>
45. Vonesch, C., Unser, M.: IEEE Trans. Image Process. **17**, 539 (2008)
46. Guerquin-Kern, M., et al.: IEEE Trans. Med. Imaging **30**, 1649 (2011)
47. Huybrechts, S.J., et al.: Traffic **10**, 1722 (2009)
48. Poole, B., Higashi, T., de Duve, C.: J. Cell Biol. **45**, 408 (1970)
49. Fujiki, Y., et al.: Peroxisome biogenesis in mammalian cells. Front Physiol. **5**, 307 (2014)
50. Kim, P.K., et al.: J. Cell Biol. **173**, 521 (2006)
51. Motley, A.M., Hettema, E.H.: J. Cell Biol. **178**, 399 (2007)
52. van der Zand, A., et al.: Cell **149**, 397 (2012)
53. Legakis, J.E.: Mol. Biol. Cell **13**(12), 4243 (2002)



Loss-Function Learning for Digital Tissue Deconvolution

Franziska Görtler¹, Stefan Solbrig², Tilo Wettig², Peter J. Oefner³,
Rainer Spang¹, and Michael Altenbuchinger¹(✉)

¹ Statistical Bioinformatics, Institute of Functional Genomics,
University of Regensburg, Am Biopark 9, 93053 Regensburg, Germany
michael.altenbuchinger@ukr.de

² Department of Physics, University of Regensburg,
Universitätsstraße 31, 93053 Regensburg, Germany

³ Institute of Functional Genomics, University of Regensburg,
Am Biopark 9, 93053 Regensburg, Germany

Abstract. The gene expression profile of a tissue averages the expression profiles of all cells in this tissue. Digital tissue deconvolution (DTD) addresses the following inverse problem: Given the expression profile y of a tissue, what is the cellular composition c of that tissue? If X is a matrix whose columns are reference profiles of individual cell types, the composition c can be computed by minimizing $\mathcal{L}(y - Xc)$ for a given loss function \mathcal{L} . Current methods use predefined all-purpose loss functions. They successfully quantify the dominating cells of a tissue, while often falling short in detecting small cell populations.

Here we use training data to learn the loss function \mathcal{L} along with the composition c . This allows us to adapt to application-specific requirements such as focusing on small cell populations or distinguishing phenotypically similar cell populations. Our method quantifies large cell fractions as accurately as existing methods and significantly improves the detection of small cell populations and the distinction of similar cell types.

1 Introduction

Different tissues of the body have different cellular compositions. The composition of tumor tissue is different from that of normal tissue. Also, when comparing two tumor tissues, their cellular composition can differ greatly. The relatively small populations of tumor-infiltrating immune cells are of particular importance. They affect progression of disease [1] and success of treatment [2]. Immune therapies block communication lines between tumor cells and infiltrating immune cells. Whether they are successful or not depends on the presence, quantity, and molecular sub-type of the infiltrating immune cells [3]. Immune-cell populations are typically small, and their molecular phenotype can be difficult to observe under the microscope. Single-cell technologies such as fluorescence-activated cell sorting (FACS; e.g. [4]), cytometry by time-of-flight (CyTOF; e.g. [5]), and single-cell RNA sequencing [6] assess molecular features on the single-cell level and can thus be used to determine the cellular tissue composition experimentally.

A more cost- and work-efficient alternative to single-cell assays is a combination of bulk-tissue gene expression profiling with digital tissue deconvolution (DTD) [7–13]. DTD addresses the following inverse problem: given the bulk gene expression profile y of a tissue, what is the cellular composition c of that tissue? Supervised DTD assumes that there is a matrix X whose columns are reference profiles of individual cell types. The composition c of y can be computed by minimizing $\mathcal{L}(y - Xc)$ for a given loss function \mathcal{L} . Competing DTD methods use different predefined all-purpose loss functions \mathcal{L} and different estimation algorithms to distil c from y and X .

The practical objective of DTD is to estimate c correctly, while the formal objective of common DTD algorithms is to estimate y correctly. If tissue expression profiles were exact mixtures of reference profiles, existing methods should work perfectly. They are not and this causes problems:

- (1) **Collections of reference profiles can be incomplete.** There might be cells in the tissue that are not represented by the reference profiles. In that case the global DTD problem is not solvable, and DTD-algorithms will compensate for the contributions of these cells by increasing the frequencies of other cell types.
- (2) **Small cell fractions are hard to quantify.** From a practical point of view this is probably the most important point, and improvements are needed badly. Immunological cell populations in a tumor are small, but they may determine the reaction of a tumor to immunotherapy. Therefore, DTD algorithms must use faint signals from small cell populations more effectively.
- (3) **Some cell types can hardly be distinguished by their expression profiles.** The profile of an epithelial cell differs greatly from that of a lymphoid cell. For two immunological sub-entities of CD8+ T cells the differences are more subtle. The more similar two cell types are, the more similar are their expression profiles, and the more difficult is their distinction.

In summary, different applications need different approaches. One way to adapt the estimation of c is to adapt the loss function \mathcal{L} . If the focus of an application is on a predefined set of cell types, genes that are informative to distinguish exactly these cells should dominate \mathcal{L} . This is even more important if the focus is on small cell populations, the faint signals of which must not be suppressed. Unfortunately, it is not clear a priori which genes to ignore and which to focus on.

2 Methods

2.1 Notations

Let $X \in \mathbb{R}^{p \times q}$ be a matrix with cellular reference profiles $X_{\cdot,j}$ in its columns, where the dot stands for all row indices. $X_{i,j}$ is the reference expression value of gene i in cells of type j , p the number of genes, and q the number of cell types in X , respectively. We further introduce a matrix $Y \in \mathbb{R}^{p \times n}$ with bulk profiles of n cell mixtures $Y_{\cdot,k}$ in its columns and a matrix $C \in \mathbb{R}^{q \times n}$ with the cellular compositions of the mixtures $C_{\cdot,k}$ as columns.

2.2 Loss-Function Learning

Following the established linear DTD algorithms, we approximate the mixture $Y_{\cdot,k}$ by a linear combination of reference profiles (the columns of X) with $C_{\cdot,k}$ as weights and estimate the composition of the k -th mixture $C_{\cdot,k}$ by minimizing

$$\mathcal{L}_g(Y_{\cdot,k} - XC_{\cdot,k}), \quad (1)$$

where

$$\mathcal{L}_g = \|\text{diag}(g)(Y_{\cdot,k} - XC_{\cdot,k})\|_2^2. \quad (2)$$

In contrast to standard DTD algorithms, which determine g by prior knowledge or separate statistical analysis, we will learn g directly from data. To this end we assume that we have a training set of mixtures $Y_{\cdot,k}$ from a specific application context with known cellular proportions $C_{\cdot,k}$ that sum to one. The entries of g are the gene weights that define the loss function. We want to learn g from the training data such that minimizing $\mathcal{L}_g(y - Xc)$ with respect to c yields accurate quantifications of cell populations for future samples with similar characteristics as those used for training.

Our method has two nested objective functions: An outer function $L(g)$ and an inner function \mathcal{L}_g , which is here given by Eq. (2). L evaluates discrepancies between the estimated and the true cellular frequencies of cell types across samples by Pearson correlation:

$$L(g) = -\sum_{j=1}^q \text{cor}(C_{j,\cdot}, \hat{C}_{j,\cdot}(g)) \quad \text{subject to } g_i \geq 0 \text{ and } \|g\|_2 = 1, \quad (3)$$

where the $\hat{C}_{j,\cdot}(g)$ are the estimates of $C_{j,\cdot}$ given g . To evaluate $L(g)$ we need to calculate all $\hat{C}_{j,\cdot}(g)$, which requires optimizing \mathcal{L}_g with respect to all $C_{\cdot,k}$. Note that if \hat{g} is a minimum of L , so is $\alpha\hat{g}$ for $\alpha > 0$. The constraint $\|g\|_2 = 1$ is thus needed to ensure unique solutions.

Note that

$$\text{cor}(C_{j,\cdot}, a_j \hat{C}_{j,\cdot}) = \text{cor}(C_{j,\cdot}, \hat{C}_{j,\cdot}), \quad (4)$$

where a_j is an arbitrary positive constant. This symmetry is important, since bulk and reference profiles must be normalized to a common mean across genes or to a common library size. A normalized reference profile $X_{\cdot,j}$ of a cell type reflects the true RNA content $\tilde{X}_{\cdot,j}$ of these cells only up to an unknown factor: $X_{\cdot,j} = \alpha_j \tilde{X}_{\cdot,j}$. Large cells with a lot of RNA have smaller α_j than smaller cells. The same is true for the bulk profiles $Y_{\cdot,k}$, where we have $Y_{\cdot,k} = \beta_k \tilde{Y}_{\cdot,k}$. The deconvolution equation

$$\tilde{Y}_{\cdot,k} = \tilde{X} \tilde{C}_{\cdot,k} + \epsilon \quad (5)$$

yields estimates \tilde{C}_{jk} that reflect the number of cells of type j . However, \tilde{Y} and \tilde{X} are not observable in practice and consequently, \tilde{C} is not accessible by DTD directly. We need to work with X and Y instead.

Note that $C_{.,k} = \tilde{C}_{.,k} / \sum_{j=1}^q \tilde{C}_{jk}$. Consider now the hypothetical deconvolution formula with normalized Y but the unobservable true \tilde{X}

$$Y_{.,k} = \tilde{X} C'_{.,k} + \epsilon. \quad (6)$$

Here, we assume $C'_{.,k} = c C_{.,k}$ for all k , where c is a positive constant. In other words we assume that if the library size of $Y_{.,k}$ is the same for all samples, we will roughly need the same number of cells to account for it. This allows us to replace \tilde{Y} by Y .

The choice of the correlation in the definition of $L(g)$ also allows us to replace \tilde{X} by X . If we write Eq. (6) using X , we obtain

$$Y_{.,k} = \sum_{j=1}^q \frac{1}{\alpha_j} X_{.,j} C'_{jk} + \epsilon. \quad (7)$$

Thus, the estimated cell frequencies are $\frac{1}{\alpha_j} C'_{j.,} = \frac{c}{\alpha_j} C_{j.,}$ and can be quite different from the training proportions $C_{j.,}$ in absolute numbers. Nevertheless, they correlate with $C_{j.,}$ and will thus generate small losses $L(g)$.

In summary, data normalization makes tissue deconvolution a non-standard deconvolution problem. The choice of correlation as loss function allows us to estimate cell frequencies independent of normalization factors.

The minimum of \mathcal{L}_g can be calculated analytically, yielding

$$\hat{C}(g) = (X^T \Gamma X)^{-1} X^T \Gamma Y \quad (8)$$

with $\Gamma = \text{diag}(g)$. Inserting this term into L leaves us with a single optimization problem in g . We minimize L by a gradient-descent algorithm. Let μ_j and σ_j be the mean and standard deviation of $C_{j.,}$, respectively. We obtain the gradient

$$\frac{\partial L(g)}{\partial g_i} = \sum_{j=1}^q \sum_{k=1}^n \frac{1}{\sigma_j \hat{\sigma}_j} \left(\frac{\text{cov}(C_{j.,}, \hat{C}_{j.,})}{n \hat{\sigma}_j^2} (\hat{C}_{jk} - \hat{\mu}_j) - \frac{1}{n} (C_{jk} - \mu_j) \right) \frac{\partial \hat{C}_{jk}(g)}{\partial g_i} \quad (9)$$

with

$$\frac{\partial \hat{C}(g)}{\partial g_i} = (X^T \Gamma X)^{-1} X^T \delta(i) (1 - X (X^T \Gamma X)^{-1} X^T \Gamma) Y, \quad (10)$$

where $\delta(i) \in \mathbb{R}^{p \times p}$ is defined as

$$\delta(i)_{jk} = \begin{cases} 1 & \text{if } i = j = k, \\ 0 & \text{else.} \end{cases} \quad (11)$$

The constraints $\|g\|_2 = 1$ and $g_i \geq 0$ were incorporated by normalizing g by its length and by restricting the search space to $g_i \geq 0$.

3 Results

3.1 DTD of Melanomas

For both training and validation we need expression profiles of cellular mixtures of known composition. We used expression data of melanomas whose composition has been experimentally resolved using single-cell RNAseq profiling [14]. These data included 4,645 single-cell profiles from 19 melanomas. The cells were annotated as T cells (2,068), B cells (515), macrophages (126), endothelial cells (65), cancer-associated fibroblasts (CAFs) (61), natural killer (NK) cells (52), and tumor/unclassified (1,758). The first 9 melanomas defined our validation cohort and the remaining 10 our training data.

First, data were transformed into transcripts per million. Then, for each cell cluster we sampled 20% of single-cell profiles in the training data, summed them up, normalized them to a common number of counts, and removed them from the training data. This yielded reference profiles $X_{\cdot,j}$. The 1,000 genes with the highest variance across all reference profiles were used to train models.

The sum of all single-cell profiles of a melanoma gave us bulk profiles. In addition, we generated a large number of artificial bulk profiles by randomly sampling single-cell profiles and summing them up. All bulk profiles were normalized to the same number of reads as those in $X_{\cdot,j}$.

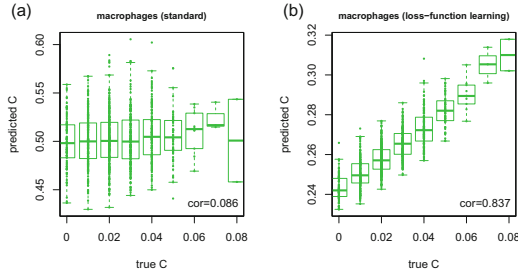


Fig. 1. Deconvolution performance with only a single reference profile (macrophages). Predicted cell frequencies are plotted versus real frequencies. Results from the standard DTD model with $g = 1$ are shown in (a), for DTD with loss-function learning in (b).

3.2 Loss-Function Learning Improves DTD Accuracy in the Case of Incomplete Reference Data

We generated 2,000 artificial cellular mixtures from our training cohort. For each of these mixtures, we randomly drew 100 single-cell profiles, summed up their raw counts, and normalized them to a fixed number of total counts. Analogously, we generated 1,000 artificial cellular validation mixtures.

Then, we restricted X to three cell types (T cells, B cells, and macrophages). Hence endothelial cells, CAFs, NK cells and tumor/unclassified cells in the mixtures are not represented in X . For standard DTD with $g = (1, \dots, 1)$,

we observed correlation coefficients of 0.70 (T cells), 0.39 (B cells), and 0.52 (macrophages) between true and estimated cell population sizes for the validation mixtures. These improved to 0.86 (T cells), 0.89 (B cells), and 0.83 (macrophages) for loss-function learning, after we ran 1000 iterations of the gradient descent algorithm on the training data. We tested our gradient descent algorithm on the 100 most variable genes for 100 different uniformly drawn starting points $g \in [0, 1]^p$. The maximal Euclidean distance between resulting composition vectors c was 2%.

To test the limits of the approach, we excluded all but the macrophages, which account for less than 3% of all cells, from the reference data X . We observed, that standard DTD broke down, while loss-function learning yielded a model that predicted macrophage abundances that still correlated well ($r = 0.84$) with the true abundances (Fig. 1).

3.3 Loss-Function Learning Improves the Quantification of Small Cell Populations

We generated data as above for mixtures of T cells, B cells, macrophages, endothelial cells, CAFs, NK cells and tumor/unclassified cells, and use all cells except the tumor cells in X . This time we control the abundance of B cells in the simulated mixtures at 0 to 5 cells, 5 to 15, 15 to 30, 30 to 50, and 50 to 75 out of 100 cells. Not surprisingly, small fractions of B cells were harder to quantify than large ones. Loss-function learning improved the accuracy for all amounts of B cells, but the improvements were greatest for small amounts (Fig. 2a). With only 0 to 5 cells in a mixture the accuracy improved from $r = 0.22$ to $r = 0.79$. Furthermore, we observed that loss-function learning on small B-cell proportions yielded a model that was highly predictive of B-cell contributions over the whole spectrum (Fig. 2a green stars).

If we compare the top-ranked genes of the model learned for the small B-cell population (Fig. 2b) to that of the macrophage-focussed simulation (Fig. 2c), we observe that the former still comprises marker genes to distinguish all cell types, while the latter focusses on genes that characterize macrophages.

3.4 Loss-Function Learning Improves the Distinction of Closely Related Cell Types

The cell types that were annotated by [14] displayed very different expression profiles. If we are interested in T-cell subtypes such as CD8+ T cells, CD4+ T-helper (Th) cells, and regulatory T cells (Tregs), reference profiles are more similar and DTD is more challenging. We subdivided the fraction of annotated T-cell profiles as follows: all T cells with positive CD8 (sum of CD8A and CD8B) and zero CD4 count were labelled CD8+ T cells (1,130). Vice versa, T cells with zero CD8 and positive CD4 count were labelled CD4+ T cells (527). These were further split into Tregs if both their FOXP3 and CD25 (IL2RA) count was positive (64), and CD4+ Th cells otherwise (463). T cells that fulfilled neither the CD4+ nor the CD8+ criteria (411) contributed to the mixtures, but were

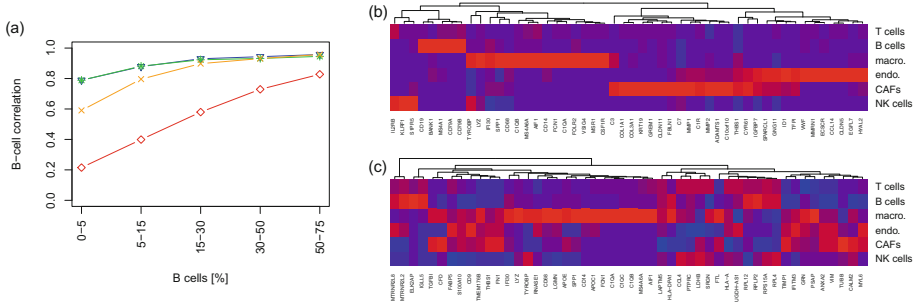


Fig. 2. Plot (a) shows how the correlation between predicted and true cellular frequencies for B cells depends on the proportion of B cells. The blue triangles correspond to models from loss-function learning and red diamonds to the standard DTD model with $g = 1$. Furthermore, the green stars show how the model trained on mixtures with 0 to 5% B cells extrapolates to higher B-cell proportions. The orange line in contrast was trained on mixtures with 50 to 75% B cells and extrapolates to lower B-cell proportions. Plot (b) shows a heatmap of the 50 most important genes corresponding to the green star model (genes were ranked by $\hat{y}_i \times \text{var}(X_{i,\cdot})$). Plot (c) shows an analogous heatmap for loss-function learning on macrophages only. Blue corresponds to low expression and red to high expression. (Color figure online)

not assessed by DTD. We augmented the reference matrix X , here consisting of T cells, B cells, macrophages, endothelial cells, CAFs and NK cells, by these cell types, replacing the original all T-cell profile with the more specific profiles for CD8+ T cells, CD4+ Th and Tregs. Then we simulated 2,000 training and 1,000 test mixtures as described above.

For standard DTD with $g = 1$ we observed correlation coefficients of 0.19 (CD4+ Th), 0.53 (CD8+), and 0.08 (Tregs) between true and estimated cell population sizes. These improved to 0.58 (CD4+ Th), 0.78 (CD8+), and 0.57 (Tregs) for our method (Fig. 3).

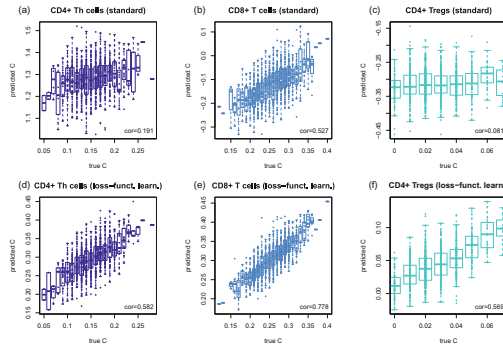


Fig. 3. Deconvolution of T-cell subentities. Results from the standard DTD model with $g = 1$ are shown in the upper row, plots (a–c), results from loss-function learning in the lower row, plots (d–e).

3.5 Loss-Function Learning is Beneficial Even for Small Training Sets, and the Performance Improves as the Training Dataset Grows

We repeated the simulation in Subsect. 3.4, but varied the size of the training dataset. We observed that loss-function learning improved accuracy for training datasets as small as 15 samples. Moreover, with more training data added the boost in performance grew and saturated only for training sets with more than 1,000 samples (Fig. 4).

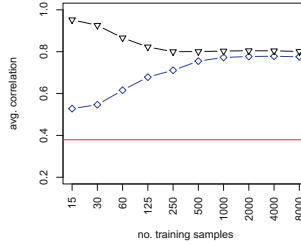


Fig. 4. Performance with and without loss-function learning as a function of the size of the training set. Performance was assessed by calculating the average correlations between predicted and true cellular contributions over all cell types. The blue diamonds and black triangles correspond to the performance of loss-function learning for the validation mixtures and training mixtures, respectively. The performance of standard DTD with $g = 1$ is shown as a red line for the validation mixtures.

3.6 HPC-Empowered Loss-Function Learning Rediscovered Established Cell Markers and Complements Them by New Discriminatory Genes for Improved Performance

Here, we introduce a final model, optimized on the 5,000 most variable genes. For this purpose, we generated 25,000 training mixtures from the melanomas of the training data. With standard desktop workstations the solution of this problem was computationally not feasible. A single computation of the gradient took 16 h (2x Intel Xeon CPU [X5650; Nehalem Six Core, 2.67 GHz], 148 Gb RAM), and this needs to be computed several hundred times until convergence. Therefore, we developed a High-Performance-Computing (HPC) implementation of our code by parallelizing Eqs. (3) and (10) with MPI, using the pbdMPI library [15, 16] as an interface. Furthermore, we linked R with the Intel Math Kernel Library for threaded and vectorized matrix operations. We ran the algorithm on 25 nodes of our QPACE 3 machine [17] with 8 MPI tasks per node and 32 hardware threads per task, where each thread can use two AVX512 vector units. In 16 h 5,086 iterations were finished, after which the loss (3) was stable to within 1%.

The high-performance model includes several genes, whose expression is characteristic for the cells distinguished in the present study. These include, among others, the CD8A gene, which encodes an integral membrane glycoprotein essential for the activation of cytotoxic T-lymphocytes [18] and the protection of a

subset of NK cells against lysis, thus enabling them in contrast to CD8- NK cells to lyse multiple target cells [19]. As evident from Fig. 5, NK cells are clearly set apart from all the other cell types studied by the expression of the killer cell lectin like receptor genes *KLRB1*, *KLRC1*, and *KLRF1* [20]. B cells, on the other hand, are clearly characterized by the expression of (i) *CD19*, which assembles with the antigen receptor of B lymphocytes and influences B-cell selection and differentiation [21], (ii) *CD20* (*MS4A1*), which is coexpressed with *CD19* and functions as a store-operated calcium channel [22], (iii) B Lymphocyte Kinase (*BLK*), a src-family protein tyrosine kinase that plays an important role in B-cell receptor signaling and phosphorylates specifically (iv) *CD79A* at Tyr-188 and Tyr-199 as well as *CD79B* (not among the top 150 genes) at Tyr-196 and Tyr-207, which are required for the surface expression and function of the B-cell antigen receptor complex [23], and (v) *BLNK*, which bridges *BLK* activation with downstream signaling pathways [24]. The expression of *FOXP3* is also highly cell specific. *FOXP3* distinguishes regulatory T cells from other CD4+ cells and functions as a master regulator of their development and function [25]. Finally, CD4+ T-helper (Th) cells are distinguished indirectly from all the other aforementioned lymphocytes by the lack of expression of cell type-specific genes. In contrast to lymphocytes, macrophages, cancer-associated fibroblasts (CAFs), and endothelial cells, which line the interior surface of blood vessels and lymphatic vessels, are characterized each by a much larger number of genes. Exemplary genes include *CD14*, *CD163*, *MSR1*, *STAB1*, and *CSF1R* for macrophages. The monocyte differentiation antigen *CD14*, for instance, mediates the innate immune response to bacterial lipopolysaccharide (LPS) by activating the NF- κ B pathway and cytokine secretion [26], while the colony stimulating factor 1 receptor (*CSF1R*) acts as a receptor for the hematopoietic growth factor *CSF1*, which controls the proliferation and function of macrophages [27]. CAFs, on the other hand, are distinguished by the expression of genes encoding extracellular matrix proteins such as fibulin-3 (*EFEMP1*), various collagens (*COL1A1*, *COL3A1*, *COL6A1*, *COL6A3*), versican (*VCAN*), a well known mediator of cell-to-cell and cell-to-matrix interactions [28] that plays critical roles in cancer biology [29], as well as the matrix metalloproteinases *MMP1* and *MMP2*, two collagen degrading enzymes that allow cancer cells to migrate out of the primary tumor to form metastases [30]. Noteworthy is also *GREM1*, an antagonist of the bone morphogenetic protein pathway. Its expression and secretion by stromal cells in tumor tissues promotes the survival and proliferation of cancer cells [31]. Genes characteristic for endothelial cells include among others *CDH5*, a member of the cadherin superfamily essential for endothelial adherens junction assembly and maintenance [32], the endothelial cell-specific chemotaxis receptor (*ECSCR*) gene, which encodes a cell-surface single-transmembrane domain glycoprotein that plays a role in endothelial cell migration, apoptosis and proliferation [33], claudin-5 (*CLDN5*), which forms the backbone of tight junction strands between endothelial cells [34], and the von Willebrand factor (*VWF*), which mediates the adhesion of platelets to sites of vascular damage by binding to specific platelet membrane glycoproteins and to constituents of exposed connective tissue [35].

We discussed 28 genes of the top 150 shown in Fig. 5. These genes have a total weight of 28% of all 5,000 gene weights (calculated as $\hat{g}_i \times \text{var}(X_{i,\cdot})$). Our algorithm complements this gene set with additional genes, including some that were, to our knowledge, not yet used to characterize cell types. An interesting example is CXorf36 (DIA1R), which has been described as being expressed at low levels in many tissues and deletion and/or mutations of which have been associated with autism spectrum disorders [36]. However, nothing is known about its function to date. Therefore, its observed overexpression in endothelial cells may provide an important clue for future study on its function.

3.7 Loss-Function Learning Shows Similar Performance as CIBERSORT for the Dominating Cell Populations and Improves Accuracy for Small Populations and in the Distinction of Closely Related Cell Types

Next we compared our model trained in Subsect. 3.6 to a competing method. For this, we generated 1,000 test mixtures from our validation melanomas. We chose CIBERSORT [12] for comparison, because it was consistently among the best DTD algorithm in a broad comparison of five different algorithms on several benchmark datasets [12]. We ran CIBERSORT on the test mixtures, using two distinct approaches: first, we uploaded our validation data to CIBERSORT using their reference profiles. The performance is summarized in Fig. 6 as CIBERSORT^a (yellow). We observed that the large population of B cells was estimated accurately, while smaller populations were inaccurate (NK cells, Tregs). Next, we uploaded our reference profiles and used the CIBERSORT gene selection (CIBERSORT^b green). We found that highly abundant cell types (B cells and CD8+ T cells) were predicted with high accuracy. However, the distinction of similar cell types such as CD4+ T helper cells and Tregs was compromised, $r = 0.42$ and $r = 0.42$, respectively. Similarly, predictions for the small populations of CAFs were compromised. That might be explained by the fact that CIBERSORT does not take into account their distinction and thus appropriate marker genes might be missing. In a direct comparison to CIBERSORT our method showed similar or better performance.

Next, we tested whether our method would have also worked for bulk profiles generated by a different technology than the reference profiles. We used the scRNAseq derived loss-function and the bulk profiles described above but replaced the reference profiles in X by microarray data downloaded from the CIBERSORT webpage. We rescaled the microarray matrix X such that the gene-wise means were identical to the scRNAseq data. Results are shown in Fig. 6 in pink. Although accuracy was slightly reduced, we still improved on the CIBERSORT results.

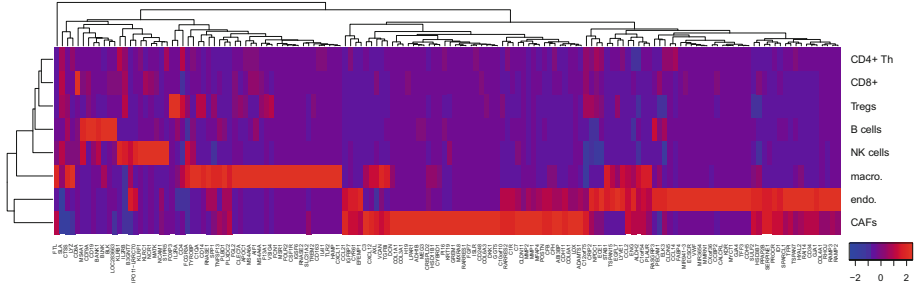


Fig. 5. Heatmap of X for the features with the top 150 weights ($\hat{g}_i \times \text{var}(X_{i,\cdot})$). Blue corresponds to low expression and red to high expression. The data were clustered by Euclidean distance. (Color figure online)

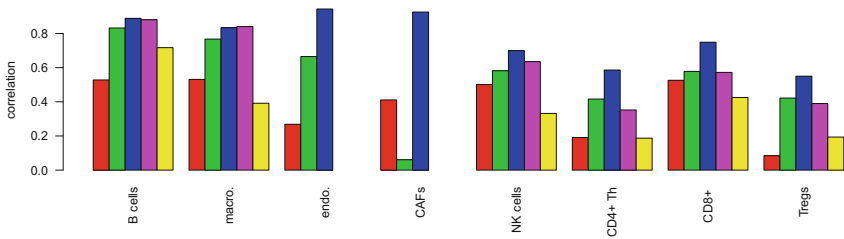


Fig. 6. Performance comparison. The methods are from left to right: standard DTD with $g = 1$ on the 5,000 most variable genes (red), CIBERSORT^b (green), loss-function learning (blue), the study where bulk and reference profiles were generated with different technologies (pink), and CIBERSORT^a (yellow). Performance was calculated as correlation between predicted and true frequencies on 1,000 validation mixtures. Endothelial cells (endo.) and CAFs were not estimated by CIBERSORT^a and microarray reference profiles were not available. Thus no yellow and pink bars are shown. (Color figure online)

3.8 Loss-Function Learning Improves the Decomposition of Bulk Melanoma Profiles

All mixtures discussed so far were artificial because only 100 single-cell profiles were chosen randomly. They might differ significantly from mixtures in real tissue. Therefore, we generated 19 full bulk melanoma profiles by summing up the respective single-cell profiles. These should reflect bulk melanomas [37]. Our predictions are contrasted with the true proportions in Fig. 7. Only the predictions for Tregs were compromised with $r = 0.48$, while the predictions for all other cell types were reliable with correlations ranging from $r = 0.70$ (CD4+ Th) to $r = 0.99$ (CAFs) on the validation melanomas.

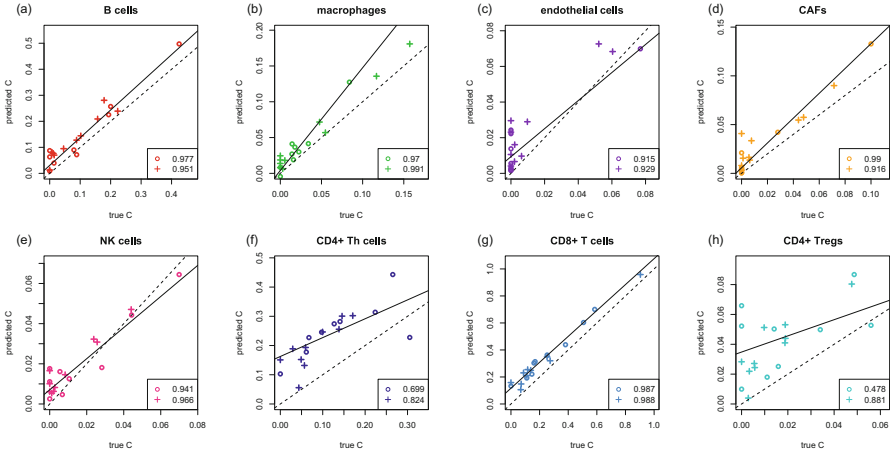


Fig. 7. Deconvolution of melanoma tissues. The circles indicate melanomas from the validation data and pluses from the training data. Figure (a) to (h) correspond to B cells, macrophages, endothelial cells, CAFs, NK cells, CD4+ Th cells, CD8+ T cells, and CD4+ Tregs, respectively. The solid black lines show the corresponding linear regression fits on the validation data, the dashed lines the identity.

4 Discussion

We suggest using training data for loss-function learning for digital tissue deconvolution to adapt the deconvolution algorithm to the requirements of specific application domains. The concept is similar to an embedded feature-selection approach in regression or classification problems. In both contexts feature selection is directly linked to a prediction algorithm and not treated as an independent preprocessing step.

The main limitation of our method is the availability of training data. Other methods do not use, and cannot use, training data. In fact, the strength of loss-function learning results primarily from the additional information in training data with known cellular compositions. Such data is not always available, but with current improvements in FACS and single-cell sequencing technology, it is becoming increasingly available.

We described and tested a specific instance of loss-function learning using squared residuals for \mathcal{L}_g . The concept is not limited to this type of inner loss function and can also be used in combination with other loss functions such as those from penalized least-squares regression [11], l_1 regression, or support vector regression [12]. However, the least-squares loss function allowed us to state the outer optimization problem in a closed analytical form, reducing computational burden.

The outer loss function L evaluates the fit of estimated and true cellular proportions in the training samples. We chose the correlation of estimated versus true quantities across samples, and no absolute measure of deviation such as

$\|c - \hat{c}\|_2^2$, which does not fulfill symmetry (4). Moreover, we did not require the estimated proportions $\hat{C}_{\cdot,k}$ for tissue k to sum up to one. Consequently, the estimated cellular composition for a given cell type is comparable between tissues, but the estimated cellular composition across cell types is not. When testing our method we did not look at absolute deviations of true versus estimated cell proportions but only at their correlation. We do not infer how many cells of a specific type (e.g., T cells) are in a tissue (Fig. 7), nor whether they constituted 10% or 20% of the cells in this tissue. However, if we had two tissues and estimated that there were more cells of that type in the first tissue compared to the second, this relation was also found in the true cell populations.

In summary, we introduced loss-function learning as a new machine-learning approach to the digital tissue deconvolution problem. It allows us to adapt to application-specific requirements such as focusing on small cell populations or delineating similar cell types. In simulations and in an application to melanoma tissues the use of training data allowed our method to quantify large cell fractions as accurately as existing methods and significantly improved the detection of small cell populations and the distinction of similar cell types.

Acknowledgement. This work was supported by BMBF (eMed Grant 031A428A) and DFG (FOR-2127 and SFB/TRR-55).

References


1. Galon, J., Costes, A., Sanchez-Cabo, F., Kirilovsky, A., Mlecnik, B., Lagorce-Pagès, C., Tosolini, M., Camus, M., Berger, A., Wind, P., et al.: Type, density, and location of immune cells within human colorectal tumors predict clinical outcome. *Science* **313**(5795), 1960–1964 (2006)
2. Fridman, W.H., Pagès, F., Sautès-Fridman, C., Galon, J.: The immune contexture in human tumours: impact on clinical outcome. *Nat. Rev. Cancer* **12**(4), 298–306 (2012)
3. Hackl, H., Charoentong, P., Finotello, F., Trajanoski, Z.: Computational genomics tools for dissecting tumour-immune cell interactions. *Nat. Rev. Genet.* **17**(8), 441–458 (2016)
4. Ibrahim, S.F., van den Engh, G.: Flow cytometry and cell sorting. In: Kumar, A., Galaev, I.Y., Mattiasson, B. (eds.) *Cell Separation. Advances in Biochemical Engineering/Biotechnology*, pp. 19–39. Springer, Heidelberg (2007). <https://doi.org/10.1007/10.2007.073>
5. Bendall, S.C., Simonds, E.F., Qiu, P., El-ad, D.A., Krutzik, P.O., Finck, R., Bruggner, R.V., Melamed, R., Trejo, A., Ornatsky, O.I., et al.: Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science* **332**(6030), 687–696 (2011)
6. Wu, A.R., Neff, N.F., Kalisky, T., Dalerba, P., Treutlein, B., Rothenberg, M.E., Mburu, F.M., Mantalas, G.L., Sim, S., Clarke, M.F., et al.: Quantitative assessment of single-cell RNA-sequencing methods. *Nat. Methods* **11**(1), 41–46 (2014)
7. Lu, P., Nakorchevskiy, A., Marcotte, E.M.: Expression deconvolution: a reinterpretation of DNA microarray data reveals dynamic changes in cell populations. *Proc. Nat. Acad. Sci. USA* **100**(18), 10370–10375 (2003)

8. Abbas, A.R., Wolslegel, K., Seshasayee, D., Modrusan, Z., Clark, H.F.: Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. *PLoS One* **4**(7), e6098 (2009)
9. Gong, T., Hartmann, N., Kohane, I.S., Brinkmann, V., Staedtler, F., Letzkus, M., Bongiovanni, S., Szustakowski, J.D.: Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples. *PLoS One* **6**(11), e27156 (2011)
10. Qiao, W., Quon, G., Csaszar, E., Yu, M., Morris, Q., Zandstra, P.W.: PERT: a method for expression deconvolution of human blood samples from varied microenvironmental and developmental conditions. *PLoS Comput. Biol.* **8**(12), e1002838 (2012)
11. Altboum, Z., Steurman, Y., David, E., Barnett-Itzhaki, Z., Valadarsky, L., Keren-Shaul, H., Meninger, T., Mendelson, E., Mandelboim, M., Gat-Viks, I., et al.: Digital cell quantification identifies global immune cell dynamics during influenza infection. *Mol. Syst. Biol.* **10**(2), 720 (2014)
12. Newman, A.M., Liu, C.L., Green, M.R., Gentles, A.J., Feng, W., Xu, Y., Hoang, C.D., Diehn, M., Alizadeh, A.A.: Robust enumeration of cell subsets from tissue expression profiles. *Nat. Methods* **12**(5), 453–457 (2015)
13. Li, B., Severson, E., Pignon, J.C., Zhao, H., Li, T., Novak, J., Jiang, P., Shen, H., Aster, J.C., Rodig, S., et al.: Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome Biol.* **17**(1), 174 (2016)
14. Tirosh, I., Izar, B., Prakadan, S.M., Wadsworth, M.H., Treacy, D., Trombetta, J.J., Rotem, A., Rodman, C., Lian, C., Murphy, G., et al.: Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**(6282), 189–196 (2016)
15. Chen, W.C., Ostrouchov, G., Schmidt, D., Patel, P., Yu, H.: pbdMPI: Programming with Big Data - Interface to MPI. R Package (2012). <https://cran.r-project.org/package=pbdMPI>
16. Chen, W.C., Ostrouchov, G., Schmidt, D., Patel, P., Yu, H.: A Quick Guide for the pbdMPI Package. R Vignette (2012). <https://cran.r-project.org/package=pbdMPI>
17. Georg, P., Richtmann, D., Wettig, T.: DD- α AMG on QPACE 3. [arXiv:1710.07041](https://arxiv.org/abs/1710.07041) (2017)
18. Veillette, A., Bookman, M.A., Horak, E.M., Bolen, J.B.: The CD4 and CD8 T cell surface antigens are associated with the internal membrane tyrosine-protein kinase p56lck. *Cell* **55**(2), 301–308 (1988)
19. Addison, E.G., North, J., Bakhsh, I., Marden, C., Haq, S., Al-Sarraj, S., Malayeri, R., Wickremasinghe, R.G., Davies, J.K., Lowdell, M.W.: Ligation of CD8 α on human natural killer cells prevents activation-induced apoptosis and enhances cytolytic activity. *Immunology* **116**(3), 354–361 (2005)
20. Moretta, A., Bottino, C., Vitale, M., Pende, D., Cantoni, C., Mingari, M.C., Biassoni, R., Moretta, L.: Activating receptors and coreceptors involved in human natural killer cell-mediated cytotoxicity. *Ann. Rev. Immunol.* **19**(1), 197–223 (2001). PMID: 11244035
21. Rickert, R.C., Rajewsky, K., Roes, J.: Impairment of T-cell-dependent B-cell responses and B-1 cell development in CD19-deficient mice. *Nature* **376**(6538), 352–355 (1995). <https://doi.org/10.1038/376352a0>
22. Li, H., Ayer, L.M., Lytton, J., Deans, J.P.: Store-operated cation entry mediated by CD20 in membrane rafts. *J. Biol. Chem.* **278**(43), 42427–42434 (2003)
23. Hsueh, R.C., Scheuermann, R.H.: Tyrosine kinase activation in the decision between growth, differentiation, and death responses initiated from the B cell antigen receptor. *Adv. Immunol.* **75**, 283–316 (2000)

24. Wienands, J., Schweikert, J., Wollscheid, B., Jumaa, H., Nielsen, P.J., Reth, M.: SLP-65: a new signaling component in B lymphocytes which requires expression of the antigen receptor for phosphorylation. *J. Exp. Med.* **188**(4), 791–795 (1998)
25. Hori, S., Nomura, T., Sakaguchi, S.: Control of regulatory T cell development by the transcription factor Foxp3. *Science* **299**(5609), 1057–1061 (2003)
26. Haziot, A., Ferrero, E., Köntgen, F., Hijiya, N., Yamamoto, S., Silver, J., Stewart, C.L., Goyert, S.M.: Resistance to endotoxin shock and reduced dissemination of gram-negative bacteria in CD14-deficient mice. *Immunity* **4**(4), 407–414 (1996)
27. Sherr, C.J., Rettenmier, C.W., Sacca, R., Rousssel, M.F., Look, A.T., Stanley, E.R.: The *c-fms* proto-oncogene product is related to the receptor for the mononuclear phagocyte growth factor, CSF 1. *Cell* **41**(3), 665–676 (1985)
28. Wu, Y.J., La Pierre, D.P., Wu, J., Yee, A.J., Yang, B.B.: The interaction of versican with its binding partners. *Cell Res.* **15**(7), 483–494 (2005)
29. Du, W., Yang, W., Yee, A.J.: Roles of versican in cancer biology - tumorigenesis, progression and metastasis. *Histol. Histopathol.* **28**(6), 701–713 (2013)
30. Gupta, A., Kaur, C.D., Jangdey, M., Saraf, S.: Matrix metalloproteinase enzymes and their naturally derived inhibitors: novel targets in photocarcinoma therapy. *Ageing Res. Rev.* **13**, 65–74 (2014)
31. Sneddon, J.B., Zhen, H.H., Montgomery, K., van de Rijn, M., Tward, A.D., West, R., Gladstone, H., Chang, H.Y., Morganroth, G.S., Oro, A.E., et al.: Bone morphogenetic protein antagonist gremlin 1 is widely expressed by cancer-associated stromal cells and can promote tumor cell proliferation. *Proc. Nat. Acad. Sci. USA* **103**(40), 14842–14847 (2006)
32. Gory-Faure, S., Prandini, M., Pointu, H., Roullot, V., Pignot-Paintrand, I., Vernet, M., Huber, P.: Role of vascular endothelial-cadherin in vascular morphogenesis. *Development* **126**(10), 2093–2102 (1999)
33. Shi, C., Lu, J., Wu, W., Ma, F., Georges, J., Huang, H., Balducci, J., Chang, Y., Huang, Y.: Endothelial cell-specific molecule 2 (ECSM2) localizes to cell-cell junctions and modulates bFGF-directed cell migration via the ERK-FAK pathway. *PLoS One* **6**(6), e21482 (2011)
34. Haseloff, R.F., Dithmer, S., Winkler, L., Wolburg, H., Blasig, I.E.: Transmembrane proteins of the tight junctions at the blood-brain barrier: structural and functional aspects. *Semin. Cell Dev. Biol.* **38**, 16–25 (2015)
35. Sadler, J.E.: Biochemistry and genetics of von Willebrand factor. *Annu. Rev. Biochem.* **67**(1), 395–424 (1998). PMID: 9759493
36. Aziz, A., Harrop, S.P., Bishop, N.E.: DIA1R is an X-linked gene related to Deleted In Autism-1. *PLoS One* **6**(1), e14534 (2011)
37. Marinov, G.K., Williams, B.A., McCue, K., Schroth, G.P., Gertz, J., Myers, R.M., Wold, B.J.: From single-cell to cell-pool transcriptomes: stochasticity in gene expression and RNA splicing. *Genome Res.* **24**(3), 496–510 (2014)



Inference of Population Structure from Ancient DNA

Tyler A. Joseph¹ and Itsik Pe'er^{1,2,3}(✉) 

¹ Department of Computer Science, Columbia University, New York, NY 10027, USA

{tjoseph,itsik}@cs.columbia.edu

² Department of Systems Biology, Columbia University, New York, NY 10027, USA

³ Data Science Institute, Columbia University, New York, NY 10027, USA

Abstract. Methods for inferring population structure from genetic information traditionally assume samples are contemporary. Yet, the increasing availability of ancient DNA sequences begs revision of this paradigm. We present Dystruct (Dynamic Structure), a framework and toolbox for inference of shared ancestry from data that include ancient DNA. By explicitly modeling population history and genetic drift as a time-series, Dystruct more accurately and realistically discovers shared ancestry from ancient and contemporary samples. Formally, we use a normal approximation of drift, which allows a novel, efficient algorithm for optimizing model parameters using stochastic variational inference. We show that Dystruct outperforms the state of the art when individuals are sampled over time, as is common in ancient DNA datasets. We further demonstrate the utility of our method on a dataset of 92 ancient samples alongside 1941 modern ones genotyped at 222755 loci. Our model tends to present modern samples as the mixtures of ancestral populations they really are, rather than the artifactual converse of presenting ancestral samples as mixtures of contemporary groups.

Keywords: Population genetics · Population structure
Ancient DNA · Time-series · Variational inference · Kalman filtering

Availability: Dystruct is implemented in C++, open-source, and available at <https://github.com/tyjo/dystruct>.

1 Introduction

The sequencing of the first ancient human genome [28], first Denisovan genome [29], and first Neanderthal genome [12] — all in 2010 — opened the floodgates for population genetic studies that include ancient DNA [19]. Ancient DNA grants a unique opportunity to investigate human evolutionary history, because it can provide direct evidence of historical relationships between populations around the world. Indeed, through combining ancient and modern samples, ancient DNA has driven many notable discoveries in human population genetics over the past ten years including the detection of introgression between anatomically modern

humans and Neanderthals [24], evidence for the genetic origin of Native Americans [26], and evidence pushing the date of human divergence in Africa to over 250,000 years ago [30], among many others [2, 9, 19, 33].

Nonetheless, incorporating new types of DNA into conventional analysis pipelines requires careful examination of existing models and tools. Ancient DNA is a particularly challenging example: individuals are sampled from multiple time points from populations where allele frequencies have drifted over time. Hence, allele frequencies are correlated over time. The current state of the art for historical inference from ancient DNA uses pairwise summary statistics calculated from genome-wide data, called drift indices or F-statistics [21, 22], not to be confused with Wright’s F-statistics, that measure the amount of shared genetic drift between pairs of populations. Drift indices have several desirable theoretical properties, such as unbiased estimators, and can be used to conduct hypothesis tests of historical relationships and admixture between sampled populations [21]. Combined with tree-building approaches from phylogenetics, drift indices can reconstruct complex population phylogenies [18] including admixture events that are robust to difference in sample times. Computing drift indices, however, requires identifying populations *a priori*, a challenging task given that multiple regions around the world experienced substantial population turnover. Thus, exploratory tools that take an unsupervised approach to historical inference are required.

One of the most ubiquitous approaches to unsupervised ancestry inference is through the Pritchard-Stephens-Donnelly (PSD) model [23], implemented in the popular software programs *structure* and ADMIXTURE [1]. Under the PSD model, sampled individuals are modeled as mixtures of latent populations, where the genotype at each locus depends on the population of origin of that locus, and allele frequencies in the latent populations. Individuals can be clustered based on their mixture proportions, the proportion of sampled loci inherited from each population, which are interpreted as estimates of global ancestry [1]. ADMIXTURE computes maximum likelihood estimates of allele frequencies and ancestry proportions under the PSD model, while *structure* uses MCMC to compute posterior expectations. A key assumption of the PSD model is that populations are in Hardy-Weinberg equilibrium: the allele frequencies in each population are fixed. For ancient DNA, this assumption is clearly violated. The robustness of the PSD model to this violation remains under-explored.

In this paper, we develop a model-based method for inferring shared history between ancient and modern samples – Dystruct (Dynamic Structure) – by extending the PSD model to time-series data. To efficiently infer model parameters, we leverage the close connection between the PSD model and another model from natural language processing: latent Dirichlet allocation (LDA) [6]. The connection between the PSD model and LDA has long been known [3, 5], and applications of the statistical methodology surrounding LDA are beginning to enter the population genetics literature [11, 27]. Similar to the PSD model, LDA models documents as mixtures of latent topics, where each topic specifies a probability distribution over words. LDA has been successfully extended to a

time-series model [5], where the word frequencies in topic distributions change over time in a process analogous to genetic drift. Thus, these dynamic topic models provide a natural starting point for models of population structure that incorporate genetic drift.

Our contributions are three-fold. First, we developed an efficient inference algorithm capable of parameter estimation under our time-series model. We extended the stochastic variational inference algorithm for the PSD model developed by [11] to time series data using the variational Kalman filtering technique developed by [5], and released software implementing our inference algorithm for general use. Second, we show that our model can lead to new insights on ancient DNA datasets: using simulations we demonstrate that Dystruct obtains more accurate ancestry estimates than ADMIXTURE on ancient DNA datasets; we then apply our model to a dataset of 92 ancient and 1941 modern samples genotyped at 222755 loci. Third, and more generally, our model opens the possibility for future model based approaches incorporating more complex demographic histories, complementing existing approaches for analyzing ancient DNA.

2 Methods

2.1 Preliminaries

Suppose we have genotypes of D individuals across L independent loci. Each individual d is a vector of L binomial observations, $\mathbf{x} = (x_{d1}, \dots, x_{dL})$ for $x_{dl} \in \{0, 1, 2\}$, where x_{dl} is the number of non-reference alleles at locus l . Each individual is assumed to have been alive during one of a finite set of time points $g[1], g[2], \dots, g[T]$. $g[t]$ is measured as number of generations since the earliest time of interest. Each individual d is time stamped by $t_d \in \{1, 2, \dots, T\}$, where $g[t_d]$ gives the time in generations when individual d was alive. We further define $\Delta g[t] = g[t] - g[t - 1]$, the time in generations between time point t and time point $t - 1$.

Under the PSD model, each individual is a mixture from K latent populations. Let $\boldsymbol{\theta}_d = (\theta_{d1}, \dots, \theta_{dK})$ be the ancestry proportions for individual d : $\boldsymbol{\theta}_d$ is the vector of probabilities that a locus in individual d originated in population k . Thus, $\sum_k \theta_{dk} = 1$. Let $\beta_{kl}[t]$ be the frequency of non-reference allele l in population k at time point t . The generative model for genotypes in each individual is (Fig. 1):

$$\boldsymbol{\theta}_d \sim \text{Dirichlet}(\alpha_1, \alpha_2, \dots, \alpha_K) \quad (1)$$

$$x_{dl} \mid \boldsymbol{\theta}_d, \beta_{1:K,l}[t_d] \sim \text{Binomial} \left(2, \sum_k \theta_{dk} \beta_{kl}[t_d] \right) \quad (2)$$

This follows the recharacterization of the original PSD model by [1] and [11].

To extend the model to time series data, we allow the allele frequencies to change at each time point using a normal approximation to genetic drift [7]:

$$\beta_{kl}[t] \mid \beta_{kl}[t - 1] \sim \text{Normal} \left(\beta_{kl}[t - 1], \frac{\Delta g[t]}{12N_k} \right) \quad (3)$$

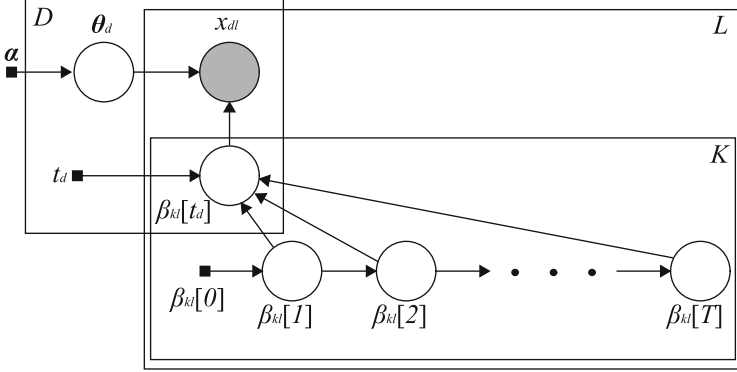


Fig. 1. Graphical model depicting Dystruct’s generative model. D individuals are genotyped at L loci from K populations (boxes), and time stamped with time point t_d . Each genotype in each individual, x_{dl} , is a binomial observation that depends on: (i) ancestry proportions, θ_d , and; (ii) allele frequencies $\beta_{kl}[t_d]$ at time point t_d . Allele frequencies $\beta_{kl}[t]$ drift over time.

N_k is the effective population size in population k . Initial allele frequencies $\beta_{kl}[0]$ and N_k are parameters of the model. Initial allele frequencies $\beta_{kl}[0]$ are estimated from data, while N_k are treated as known and fixed.

The state space model here is slightly different than normal approximation to the Wright-Fisher model for genetic drift. Under the Wright-Fisher model, the variance of allele frequencies $\Delta g[t]$ generations in the future, given the current allele frequency $\beta_{kl}[t-1]$, is $\frac{\Delta g_t \beta_{kl}[t-1](1-\beta_{kl}[t-1])}{2N_k}$.

We approximate the variance by averaging over the interval $(0, 1)$:

$$\int_0^1 \frac{\Delta g[t] \beta_{kl}[t-1](1-\beta_{kl}[t-1])}{2N_k} d\beta_{kl}[t-1] = \frac{\Delta g[t]}{12N_k} \quad (4)$$

In practice, through simulations, we found that we were able to obtain accurate estimates despite this approximation.

2.2 Posterior Inference

We take a Bayesian approach by inferring ancestry proportions through the posterior distribution $p(\theta_{1:D}, \beta_{1:K, 1:L} | \mathbf{x}_{1:D, 1:L})$. Direct posterior inference is intractable because the normal distribution is not a conjugate prior for the binomial. Following [5], we derive a variational inference algorithm that approximates the true posterior. We hereby summarize the variational inference approach for completion.

Variational inference methods [4, 15, 34] approximate the true posterior by specifying a computationally tractable family of approximate posterior distributions indexed by variational parameters, ϕ . These parameters are then optimized to minimize the Kullback-Leibler (KL) divergence between the true posterior and

its variational approximation. The key to variational inference algorithms relies on the observation that, given some distribution of latent parameters $q(\mathbf{z})$, the log likelihood of the observations \mathbf{x} can be decomposed into two terms:

$$\log p_{\boldsymbol{\eta}}(\mathbf{x}) = \int \log \left(\frac{p_{\boldsymbol{\eta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x})} \frac{q_{\boldsymbol{\phi}}(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})} \right) q_{\boldsymbol{\phi}}(\mathbf{z}) d\mathbf{z} \quad (5)$$

$$= \mathbb{E}_q \left[\log \left(\frac{p_{\boldsymbol{\eta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z})} \right) \right] + \mathbb{E}_q \left[\log \left(\frac{q_{\boldsymbol{\phi}}(\mathbf{z})}{p_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x})} \right) \right] \quad (6)$$

$$= L(\boldsymbol{\eta}, \boldsymbol{\phi}; \mathbf{x}) + \text{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}) || p_{\boldsymbol{\eta}}(\mathbf{z}|\mathbf{x})) \quad (7)$$

where $\boldsymbol{\eta}$ are the model parameters and $\boldsymbol{\phi}$ are the variational parameters. The term on the right is the KL divergence between the true posterior and the variational approximation. Because the KL divergence is non-negative, the term L , the evidence lower bound (ELBO), is a lower bound on the log-likelihood. In practice L is maximized, thereby minimizing the KL divergence between the true and approximate posterior.

We approximate the true posterior of our model with the variational posterior

$$q(\boldsymbol{\beta}_{1:K, 1:L}, \boldsymbol{\theta}_{1:D}) = \prod_{d=1}^D q(\boldsymbol{\theta}_d | \hat{\boldsymbol{\theta}}_d) \prod_{k=1}^K \prod_{l=1}^L \prod_{t=1}^T q(\beta_{kl}[t] | \hat{\beta}_{kl}[1:T]) \quad (8)$$

$q(\boldsymbol{\theta}_d; \hat{\boldsymbol{\theta}}_d)$ specifies a Dirichlet($\hat{\boldsymbol{\theta}}_d$) distribution. In the next section we elaborate on the form of $q(\beta_{kl}[t] | \hat{\beta}_{kl}[1:T])$.

2.3 Variational Kalman Filtering

Successful variational inference algorithms depend on formulating an approximate posterior close in form to the true posterior, and such that the expectations that make up the ELBO are tractable. To maintain the relationship between the relationship between the $\beta_{kl}[t]$ over time, we use Variational Kalman filtering, developed by [5] for inference in state space models with intractable posteriors. Variational Kalman filtering introduces variational parameters $\hat{\beta}_{kl}[t]$ that are pseudo-observations from the state space model:

$$\hat{\beta}_{kl}[t] | \beta_{kl}[t] \sim \text{Normal}(\beta_{kl}[t], \nu^2) \quad (9)$$

ν is an additional variational parameter. Given the pseudo-observations, standard Kalman filtering and smoothing equations can be used to calculate marginal means, $\tilde{m}_{kl}[t]$, and marginal variances, $\tilde{v}_{kl}[t]$, of the latent variables $\beta_{kl}[1:T]$ given the pseudo-observations $\hat{\beta}_{kl}[1:T]$. The variational approximation takes the form

$$\beta_{kl}[t] | \hat{\beta}_{kl}[1:T] \sim \text{Normal}(\tilde{m}_{kl}[t], \tilde{v}_{kl}[t]) \quad (10)$$

The ELBO is maximized with respect to the pseudo-observations using a conjugate gradient algorithm for numerical optimization.

2.4 Stochastic Variational Inference

Variational inference algorithms in the setting above often rely on optimizing parameters through coordinate ascent: each parameter is updated iteratively while the others remain fixed. Coordinate ascent can be computationally expensive, especially as the size of the data becomes large. Instead, we optimize the ELBO using stochastic variational inference [4, 14]. Briefly, stochastic variational inference distinguishes global variational parameters, such as $\hat{\theta}_d$, whose coordinate ascent update requires iterating through the entire dataset, with local parameters, $\hat{\beta}_{kl}$, whose update only depends on a subset of the data. We first subsample a particular locus l , update the pseudo-outputs for that locus, then update the variational parameters $\hat{\theta}_d$ by taking a weighted average of the previous parameter estimates with an estimate obtained using locus l alone. This process continues until the $\hat{\theta}_d$ converge. Estimates of ancestry proportions are computed by taking the posterior expectation of $\theta_d : \mathbb{E}_q[\theta_{dk}] = \frac{\hat{\theta}_{dk}}{\sum_s \hat{\theta}_{ds}}$.

We further optimized our implementation, obtaining an order of magnitude speed up over a naive implementation. This improvement makes Dystruct feasible to use on realistic size datasets (see Sect. 3.3).

2.5 Simulated Data

We designed simulations to test the ability of our method to assign ancient samples into populations under two historical scenarios (Fig. 2). In each scenario, we simulated K populations at 10000 independent loci according to the Wright-Fisher model for genetic drift. We drew initial allele frequencies from a Uniform(0.2, 0.8) distribution, and simulated discrete generations by drawing $2N_k$ individuals randomly with replacement from the previous generation. When then drew individuals at specific time points with genotypes and ancestry proportions specified by the generative model based on the allele frequencies at that time point. Note that we are not simulating data under the normal approximation. We fixed effective population to $N_k = 2500$ for all $k = 1, \dots, K$. To generalize our results across different effective population sizes, we measured time in coalescent units (1 coalescent unit = $2N_k$ generations). We denote the total simulation time in coalescent units by τ . Each simulation was run across $\tau \in \{0.02, 0.04, 0.08, 0.16\}$.

One concern is that our model assumes allele frequencies are away from 0 or 1, while the allele frequencies in the Wright-Fisher model are guaranteed to fix given sufficient time. We allowed allele frequencies to fix in our simulations to test our model’s robustness to violating this assumption, though most allele frequencies do not reach fixation.

In the *baseline* simulation scenario (Fig. 2a), we sampled 40 individuals from $K = 3$ populations at 3 evenly spaced time points. We drew ancestry proportions from a Dirichlet($\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$) distribution, ensuring that the majority of any one individual’s genome originated in a single population, with smaller ancestry proportions from the remaining populations.

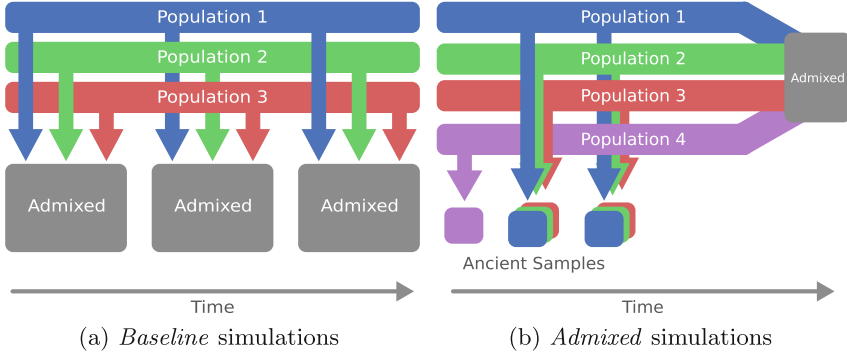


Fig. 2. Simulation scenarios explored with Dystruct. (a) *Baseline* simulation scenario. Three populations were simulated that admixed at three time points. Individuals were sampled from the admixed populations. (b) *Admixed* simulation scenario. Ancient individuals were sampled from four source populations that merged to form a modern admixed population. Modern samples were from the admixed population.

In the *admixed* scenario (Fig. 2b) we performed simulations that try to better mimic available real data. We assumed a modern population resulted from the instantaneous admixture of $K = 4$ ancestral populations. Ancient individuals were sampled pre-admixture and modern individuals were sampled post-admixture. We included two additional features found in current datasets. Such datasets comprise a small number of ancient samples when compared with modern samples. We therefore simulated 508 samples where 23 of the samples were ancient and the remaining 485 were modern, reflecting the $\sim 1:21$ balance of samples in Sect. 2.6. All ancient samples occurred before time $\frac{\tau}{2}$. One of the four ancient populations was observed in the oldest ancient sample only, but appeared in modern populations, reflecting the possibility that an ancient population may only be sampled once.

We repeated each simulation scenario 10 times for a total of 80 simulations, and compared the ability of our model to infer the parameter θ_d with that of ADMIXTURE (v1.3.0). Since effective population size is a fixed parameter in Dystruct, we tested Dystruct on several effective population sizes. We ran Dystruct with $N_k = 1000, 2500, 5000, 10000$ for all simulation scenarios. For each simulation, we computed the root-mean-square error (RMSE) between the true ancestry proportions, and parameters inferred by Dystruct and ADMIXTURE:
$$\text{RMSE}(\theta^{true}, \theta^{inf}) = \sqrt{\frac{1}{DK} \sum_{d=1}^D \sum_{k=1}^K (\theta_{dk}^{true} - \theta_{dk}^{inf})^2}.$$

2.6 Real Data

[13] analyze a hybrid dataset of modern humans from the Human Origins dataset [17, 21], 69 newly sequenced ancient Europeans, along with 25 previously published ancient samples [10, 32], to study population turnover in Europe. Ancient samples included several Holocene hunter gatherers ($\sim 5-6$ thousand

years ago; kya), Neolithic farmers (~ 5 – 8 kya), Copper/Bronze age individuals (~ 3.1 – 5.3 kya), and an Iron Age individual (~ 2.9 kya). In addition, the data include three Pleistocene hunter-gatherers — ~ 45 kya Ust-Ishim [8], ~ 30 kya Kostenki14 [31], and ~ 24 kya MA1 [26] — the Tyrolean Iceman [16], the hunter-gatherers LaBran1 [20] and Loschbour [17], and the Neolithic farmer Stuttgart [17].

We analyzed the publicly available dataset from <https://reich.hms.harvard.edu/datasets>. After removing related individuals identified in [13], and removing samples from outside the scope of their paper, we were left with a dataset consisting of 92 ancient samples and 1941 modern samples genotyped at 354212 loci. Again following [13], we pruned this original dataset for linkage disequilibrium in PLINK [25] (v1.07) using `--indep-pairwise 200 25 0.5`, leaving 222755 SNPs. To convert radiocarbon dates to generation time required by Dystruct, we assumed a 25 year generation time, and took the midpoint of the radiocarbon dates as point estimates divided by 25 for ancient samples. We further grouped time points for samples together if they were within the 95 % confidence interval for radiocarbon date estimates, and were part of the same culture. The final dataset spanned 1800 generations.

We then ran ADMIXTURE and Dystruct on the full data with effective population size of 7500 from $K = 2$ to $K = 16$ – the best supported K in [13] – and compared the results. Here we report the results for $K = 11$ because they have the clearest historical interpretation.

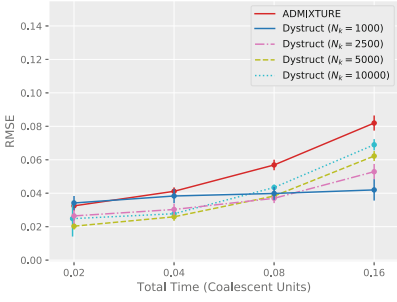
3 Results

3.1 Simulated Data

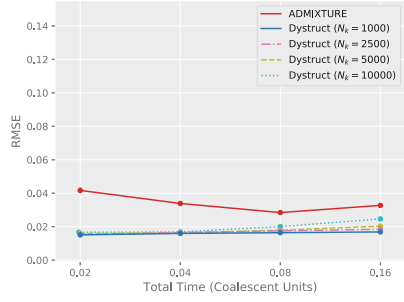
When simulating data according to the *baseline* scenario, Dystruct consistently matches up with ADMIXTURE or significantly outperforms it (Fig. 3a). When interpreting the order of magnitude of these accuracy results, it is important to note that ancestry vectors sum to 1, so a 0.01 decrease in RMSE improves relative accuracy of these vectors by order of $K\%$. ADMIXTURE performs much worse as the simulated coalescent time increases, from RMSE of 0.032 for $\tau = 0.02$ to RMSE of 0.082 at $\tau = 0.16$. Dystruct is less susceptible to this increase in error. Intuitively, the more coalescent time is considered, the more the drift, and hence, the more important it is to model its dynamics.

The *admixed* simulation scenario demonstrates a substantial advantage to Dystruct on ancient samples across population parameters (Fig. 4a). Nonetheless a near zero RMSE for Dystruct is potentially misleading because ancient samples are not admixed.

Dystruct also performs well across all samples (Fig. 3b), and on modern samples only (Fig. 4b). On modern samples, Dystruct outperforms ADMIXTURE for $\tau = 0.02, 0.04$ by a factor of 2. At $\tau = 0.08$, RMSE for ADMIXTURE and Dystruct are similar, while ADMIXTURE has a slight advantage at $\tau = 0.16$.

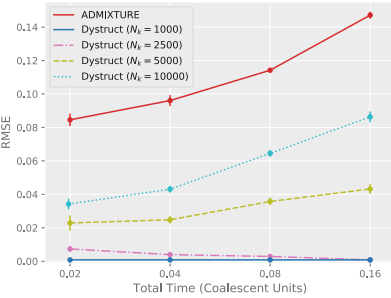


(a) RMSE for *baseline* simulations

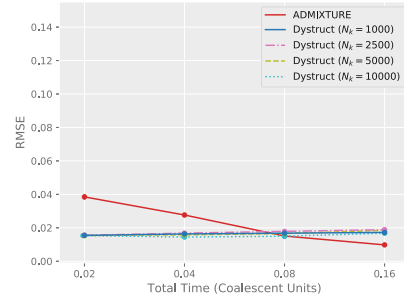


(b) RMSE for *admixed* simulations

Fig. 3. RMSE for the (a) *baseline* and (b) *admixed* simulation scenarios. Dystruct outperforms ADMIXTURE across several population size parameters for both scenarios. Ancestry vectors sum to 1, so a 0.01 improvement in RMSE corresponds to a $K\%$ performance improvement.



(a) RMSE for ancient samples



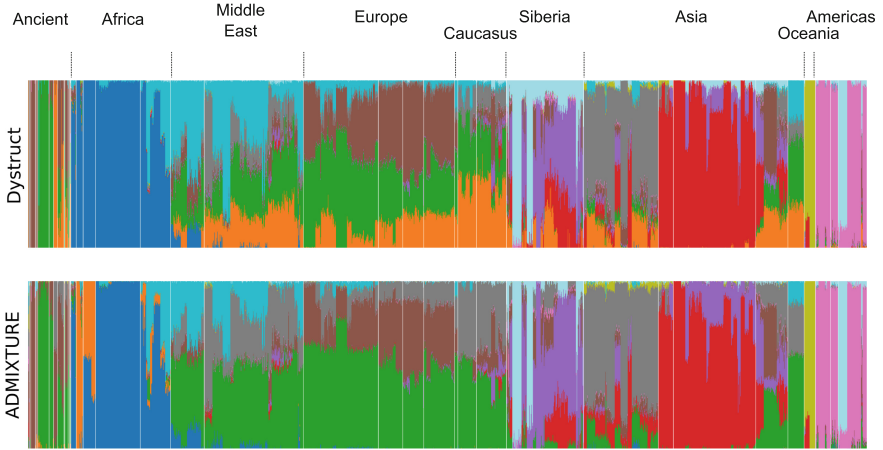
(b) RMSE for modern samples

Fig. 4. RMSE for ancestry estimates for (a) ancient samples and (b) modern samples for the *admixed* simulation scenario. Dystruct significantly outperforms ADMIXTURE when ancient samples are unadmixed (minimum RMSE = 0.00083). On modern samples, the error remains low for both Dystruct and ADMIXTURE.

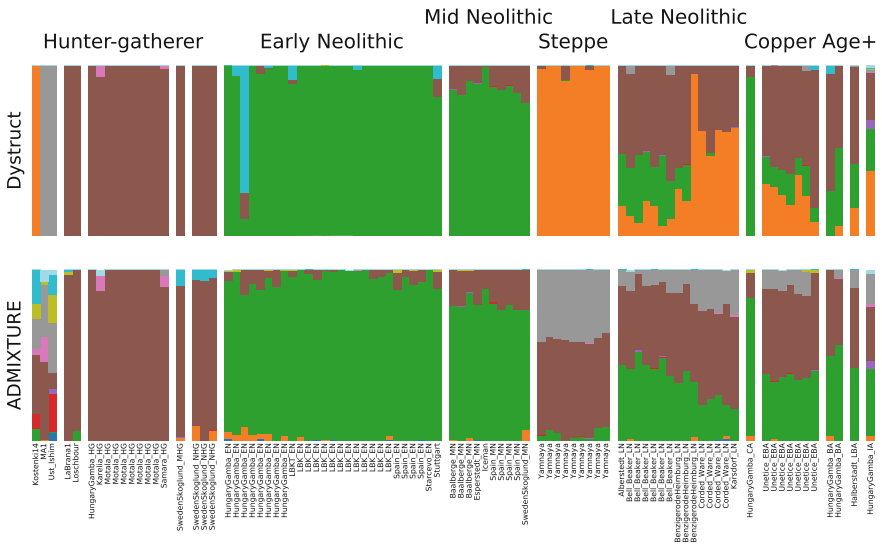
3.2 Real Data

Dystruct shows good concordance with ADMIXTURE on modern data with known global populations (Fig. 5a). In particular, African populations (Dark Blue; eg. Bantu, Mbuti, Yoruba), Asian populations (Red; e.g. Han, Japanese, Korean), Native American populations (Dark Pink; e.g. Mixe, Mayan, Zapotec), and Oceanian populations (Yellow; e.g. Papuan) all form similar genetic clusters, among many other examples.

Dystruct and ADMIXTURE differ on the ancient samples. In Dystruct, most ancient samples are “pure,” containing ancestry components from a single population, and modern day populations appear as mixtures of ancient populations. This is evident in the entropy across samples (Fig. 6). On ancient samples, Dystruct has lower entropy than ADMIXTURE, while the opposite is true for mod-



(a) Ancestry proportions inferred by Dystruct and ADMIXTURE across all samples. ADMIXTURE and Dystruct agree on several major population clusters, but differ on modern day ancestry estimates from ancient samples.



(b) Ancestry estimates for 92 ancient samples. The three leftmost samples are the Pleistocene hunter-gatherers. In Dystruct, late Neolithic samples and beyond present as a mixture of hunter-gatherers, Yamnaya steppe herders, and early Neolithic samples, matching supported historical migrations of steppe herders into Eastern and Western Europe.

Fig. 5. Ancestry proportions inferred across (a) all samples and (b) ancient samples only. Colors correspond between (a) and (b). Dystruct estimates ancestry for modern populations as combinations of ancient samples, while ADMIXTURE estimates ancestry for ancient samples as combinations of modern populations. (Color figure online)

ern samples. This is most apparent in the different ancestry assignments for the oldest samples: the Pleistocene hunter gatherers. MA1, Kostenki14, and Ust-Ishim differ substantially in their representation between Dystruct and ADMIXTURE. These are the samples where genetic drift is most prominent. ADMIXTURE analysis describes MA1, Kostenki14, and Ust-Ishim as mixtures of several modern day populations. In contrast, Dystruct describes modern populations as mixtures of components derived from MA1, Kostenki14, and Ust-Ishim.

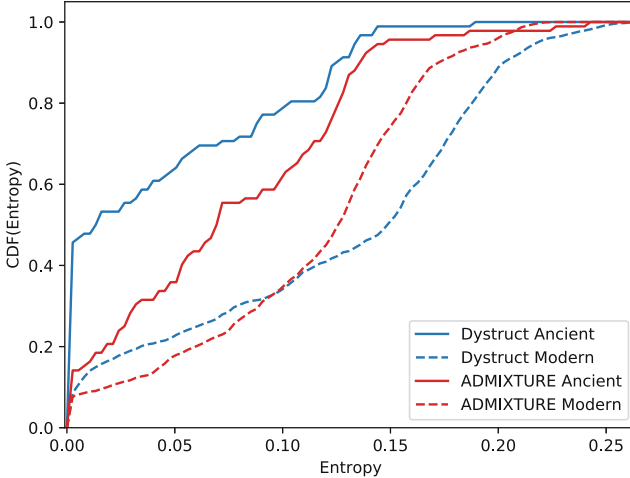


Fig. 6. Cumulative density function for entropy across ancient and modern samples. Dystruct has a lower entropy for ancient samples, while ADMIXTURE has a lower entropy for modern samples.

Most interestingly, the later ancient samples appear as mixtures of earlier samples in Dystruct, but not in ADMIXTURE. Late Neolithic, Bronze Age, and Iron Age samples appear as admixed between Yamnaya steppe herders (Orange), hunter-gatherers (Brown), and early Neolithic (Green). Additionally, we see substantial shared ancestry between these groups and modern European populations. Both findings are consistent with [13], who found evidence supporting migration out of Yamnaya steppe herders into Eastern and Western Europe ~ 4.5 kya, and supporting a model of European populations as a mixture of these groups. Kostenki14 shares ancestry with the Yamnaya group, suggesting a possible source for Yamnaya steppe ancestry.

3.3 Running Time

Despite the added complexity, additional model parameters, and large dataset, Dystruct ran on the real data in approximately 6 days using 2 cores of a 2.9 GHz Intel Core i5 processor. ADMIXTURE ran in approximately 2 days. Dystruct

ran in 30 and 120 m per replicate per core for the *baseline* and *admixed* scenarios respectively. ADMIXTURE ran in less than a minute.

Dystruct’s main computational consideration is the number of time points. During each iteration the parameters of a single locus are updated, then used to update ancestry estimates across all individuals. Estimates for ancestry parameters, $\hat{\theta}_d$, can be computed in closed form in $O(DK)$; however, the update for the parameters $\hat{\beta}_{kl}$ is approximated numerically. Computing the gradient of the $\hat{\beta}_{kl}$ at a locus takes $O(T^2 + D)$ time because the marginal means $\tilde{m}_{kl}[t]$ must be differentiated with respect to each pseudo-output.

4 Discussion

We have presented Dystruct, a model and inference procedure to understand population structure and admixture from ancient DNA. The novelty of the model is its explicit temporal semantics. This formalization of allele frequency dynamics facilitates perception of modern and more recent populations as evolved from more ancient ones or combinations thereof. We derived an efficient inference algorithm for the model parameters using stochastic variational inference, and released software for use by the broader community. We established the performance of our model on several simulation scenarios, and further demonstrated its utility for gaining insight from the analysis of real data.

Our model outperforms the current standard modeling across a variety simulation scenarios. Encouragingly, our simulations show that Dystruct does a better job recovering population structure in the presence of genetic drift, an effect that hinders existing tools. Our model accurately detects when modern populations are mixtures of pure ancestral samples, while ADMIXTURE does not, and therefore is useful for testing hypotheses of historical admixture between ancient and modern populations.

We note the advantage of Dystruct increases with genetic drift and thus with coalescent time elapsed. This means that in practical situations, where samples are dated in years, Dystruct is most important when the effective population sizes are small. From statistical inference perspective, effective population size can be thought of as a regularizer that penalizes the difference between allele frequencies at each time point. Thus, as effective population size increases, alleles frequencies drift more slowly and become closer across time points, and estimates more closely match that of ADMIXTURE.

Our results on real data match known population clusters on modern populations, and lead to new interpretations of the ancient dataset. Interestingly, the PSD model tends to describe the oldest ancient samples as mixtures of modern populations, while in Dystruct several modern populations appeared as mixtures of these ancient samples. This makes sense in light of the standard goal of maximizing overall variance explained, a quantity dominated by the majority of the samples, which are modern. In contrast, temporal semantics implicitly assume admixture occurs forward in time, putting the focus on ancient populations. Dystruct can thus provide additional insight into such populations from ancient DNA.

There are several limitations to our approach. First, we model populations as independently evolving over time. This ignores historical relationships such as population splits. Hence, Dystruct may potentially only capture one branch of a population phylogeny at a time. Second, across all simulations and for real data we constrained the effective population size across all populations to be the same. Thus, the parameters converge to one of at least K symmetric modes — population labels are exchangeable — and it is unclear how allowing different effective population sizes for different populations changes the log likelihood with respect to the parameter space. Future work should investigate this issue in more detail. Nonetheless, as we have demonstrated this is not a serious limitation to achieving reasonable estimates. Our results hold across a range of effective population sizes provided to Dystruct. Third, there is no clear procedure for choosing the optimal number of populations K . We have deferred this issue to future work, but pose that this does not prevent a severe limitation: the current state of the art uses runs across multiple values of K , and interprets the results for each K .

More generally, we have presented a time-series model for population history with several promising extensions. Our method complements existing approaches, and can lead to new insights on ancient DNA datasets. Our work represents a first step toward statistical models capable of detecting complex population histories.

Acknowledgements. This material is based upon work supported by the National Science Foundation (NSF) Graduate Research Fellowship under Grant No. DGE 16-44869, and the NSF under Grant No. DGE-1144854, and Grant No. CCF 1547120. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the NSF.

References

1. Alexander, D.H., Novembre, J., Lange, K.: Fast model-based estimation of ancestry in unrelated individuals. *Genome Res.* **19**(9), 1655–1664 (2009)
2. Allentoft, M.E., Sikora, M., Sjögren, K.G., Rasmussen, S., Rasmussen, M., Stenderup, J., Damgaard, P.B., Schroeder, H., Ahlström, T., Vinner, L., et al.: Population genomics of bronze age Eurasia. *Nature* **522**(7555), 167–172 (2015)
3. Blei, D.M.: Probabilistic topic models. *Commun. ACM.* **55**(4), 77–84 (2012)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**(518), 859–877 (2017)
5. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: *Proceedings of the International Conference on Machine Learning*, pp. 113–120. ACM (2006)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
7. Cavalli-Sforza, L.L., Edwards, A.W.: Phylogenetic analysis: models and estimation procedures. *Evolution* **21**(3), 550–570 (1967)
8. Fu, Q., Li, H., Moorjani, P., Jay, F., Slepchenko, S.M., Bondarev, A.A., Johnson, P.L., Aximu-Petri, A., Prüfer, K., de Filippo, C., et al.: Genome sequence of a 45,000-year-old modern human from western Siberia. *Nature* **514**(7523), 445–449 (2014)

9. Fu, Q., Posth, C., Hajdinjak, M., Petr, M., Mallick, S., Fernandes, D., Furtwängler, A., Haak, W., Meyer, M., Mittnik, A., et al.: The genetic history of ice age Europe. *Nature* **534**, 200 (2016)
10. Gamba, C., Jones, E.R., Teasdale, M.D., McLaughlin, R.L., Gonzalez-Fortes, G., Mattiangeli, V., Domboróczki, L., Kóvári, I., Pap, I., Anders, A., et al.: Genome flux and stasis in a five millennium transect of European prehistory. *Nat. Commun.* **5**, 5257 (2014)
11. Gopalan, P., Hao, W., Blei, D.M., Storey, J.D.: Scaling probabilistic models of genetic variation to millions of humans. *Nat. Genet.* **48**(12), 1587 (2016)
12. Green, R.E., Krause, J., Briggs, A.W., Maricic, T., Stenzel, U., Kircher, M., Patterson, N., Li, H., Zhai, W., Fritz, M.H.Y., et al.: A draft sequence of the neandertal genome. *Science* **328**(5979), 710–722 (2010)
13. Haak, W., Lazaridis, I., Patterson, N., Rohland, N., Mallick, S., Llamas, B., Brandt, G., Nordenfelt, S., Harney, E., Stewardson, K., et al.: Massive migration from the steppe was a source for Indo-European languages in Europe. *Nature* **522**(7555), 207–211 (2015)
14. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.W.: Stochastic variational inference. *J. Mach. Learn. Res.* **14**(1), 1303–1347 (2013)
15. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. *Mach. Learn.* **37**(2), 183–233 (1999)
16. Keller, A., Graefen, A., Ball, M., Matzas, M., Boisguerin, V., Maixner, F., Leiding, P., Backes, C., Khairat, R., Forster, M., et al.: New insights into the Tyrolean Iceman’s origin and phenotype as inferred by whole-genome sequencing. *Nat. Commun.* **3**, 698 (2012)
17. Lazaridis, I., Patterson, N., Mittnik, A., Renaud, G., Mallick, S., Kirsanow, K., Sudmant, P.H., Schraiber, J.G., Castellano, S., Lipson, M., et al.: Ancient human genomes suggest three ancestral populations for present-day Europeans. *Nature* **513**(7518), 409–413 (2014)
18. Lipson, M., Loh, P.R., Levin, A., Reich, D., Patterson, N., Berger, B.: Efficient moment-based inference of admixture parameters and sources of gene flow. *Mol. Biol. Evol.* **30**(8), 1788–1802 (2013)
19. Nielsen, R., Akey, J.M., Jakobsson, M., Pritchard, J.K., Tishkoff, S., Willerslev, E.: Tracing the peopling of the world through genomics. *Nature* **541**(7637), 302–310 (2017)
20. Olalde, I., Allentoft, M.E., Sánchez-Quinto, F., Santpere, G., Chiang, C.W., DeGiorgio, M., Prado-Martinez, J., Rodríguez, J.A., Rasmussen, S., Quilez, J., et al.: Derived immune and ancestral pigmentation alleles in a 7,000-year-old Mesolithic European. *Nature* **507**(7491), 225–228 (2014)
21. Patterson, N., Moorjani, P., Luo, Y., Mallick, S., Rohland, N., Zhan, Y., Genschoreck, T., Webster, T., Reich, D.: Ancient admixture in human history. *Genetics* **192**(3), 1065–1093 (2012)
22. Peter, B.M.: Admixture, population structure, and F-statistics. *Genetics* **202**(4), 1485–1501 (2016)
23. Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. *Genetics* **155**(2), 945–959 (2000)
24. Prüfer, K., Racimo, F., Patterson, N., Jay, F., Sankararaman, S., Sawyer, S., Heinze, A., Renaud, G., Sudmant, P.H., De Filippo, C., et al.: The complete genome sequence of a Neanderthal from the Altai Mountains. *Nature* **505**(7481), 43–49 (2014)

25. Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A., Bender, D., Maller, J., Sklar, P., De Bakker, P.I., Daly, M.J., et al.: PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81**(3), 559–575 (2007)
26. Raghavan, M., Skoglund, P., Graf, K.E., Metspalu, M., Albrechtsen, A., Moltke, I., Rasmussen, S., Stafford Jr., T.W., Orlando, L., Metspalu, E., et al.: Upper Palaeolithic Siberian genome reveals dual ancestry of Native Americans. *Nature* **505**(7481), 87–91 (2014)
27. Raj, A., Stephens, M., Pritchard, J.K.: fastSTRUCTURE: variational inference of population structure in large SNP data sets. *Genetics* **197**(2), 573–589 (2014)
28. Rasmussen, M., Li, Y., Lindgreen, S., Pedersen, J.S., Albrechtsen, A., Moltke, I., Metspalu, M., Metspalu, E., Kivisild, T., Gupta, R., et al.: Ancient human genome sequence of an extinct Palaeo-Eskimo. *Nature* **463**(7282), 757–762 (2010)
29. Reich, D., Green, R.E., Kircher, M., Krause, J., Patterson, N., Durand, E.Y., Viola, B., Briggs, A.W., Stenzel, U., Johnson, P.L., et al.: Genetic history of an archaic hominin group from Denisova Cave in Siberia. *Nature* **468**(7327), 1053–1060 (2010)
30. Schlebusch, C.M., Malmström, H., Günther, T., Sjödin, P., Coutinho, A., Edlund, H., Munters, A.R., Vicente, M., Steyn, M., Soodyall, H., et al.: Southern African ancient genomes estimate modern human divergence to 350,000 to 260,000 years ago. *Science* **358**(6383), 652–655 (2017)
31. Seguin-Orlando, A., Korneliussen, T.S., Sikora, M., Malaspinas, A.S., Manica, A., Moltke, I., Albrechtsen, A., Ko, A., Margaryan, A., Moiseyev, V., et al.: Genomic structure in Europeans dating back at least 36,200 years. *Science* **346**(6213), 1113–1118 (2014)
32. Skoglund, P., Malmström, H., Omrak, A., Raghavan, M., Valdiosera, C., Günther, T., Hall, P., Tambets, K., Parik, J., Sjögren, K.G., et al.: Genomic diversity and admixture differs for stone-age Scandinavian foragers and farmers. *Science* **344**(6185), 747–750 (2014)
33. Skoglund, P., Malmström, H., Raghavan, M., Storå, J., Hall, P., Willerslev, E., Gilbert, M.T.P., Götherström, A., Jakobsson, M.: Origins and genetic legacy of Neolithic farmers and hunter-gatherers in Europe. *Science* **336**(6080), 466–469 (2012)
34. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.* **1**(1–2), 1–305 (2008)



Using Minimum Path Cover to Boost Dynamic Programming on DAGs: Co-linear Chaining Extended

Anna Kuosmanen¹, Topi Paavilainen¹, Travis Gagie², Rayan Chikhi³,
Alexandru Tomescu¹, and Veli Mäkinen¹(✉)

¹ Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki, Helsinki, Finland
veli.makinen@helsinki.fi

² Diego Portales University, Santiago, Chile

³ CNRS, CRISAL, University of Lille 1, Villeneuve-d'Ascq, France

Abstract. Aligning sequencing reads on graph representations of genomes is an important ingredient of *pan-genomics*. Such approaches typically find a set of *local anchors* that indicate plausible matches between substrings of a read to subpaths of the graph. These anchor matches are then combined to form a (semi-local) alignment of the complete read on a subpath. *Co-linear chaining* is an algorithmically rigorous approach to combine the anchors. It is a well-known approach for the case of two sequences as inputs. Here we extend the approach so that one of the inputs can be a directed acyclic graph (DAGs), e.g. a *splicing graph* in transcriptomics or a *variant graph* in pan-genomics.

This extension to DAGs turns out to have a tight connection to the *minimum path cover* problem, asking us to find a minimum-cardinality set of paths that cover all the nodes of a DAG. We study the case when the size k of a minimum path cover is small, which is often the case in practice. First, we propose an algorithm for finding a minimum path cover of a DAG (V, E) in $O(k|E|\log|V|)$ time, improving all known time-bounds when k is small and the DAG is not too dense. Second, we introduce a general technique for extending dynamic programming (DP) algorithms from sequences to DAGs. This is enabled by our minimum path cover algorithm, and works by mimicking the DP algorithm for sequences on each path of the minimum path cover. This technique generally produces algorithms that are slower than their counterparts on sequences only by a factor k . Our technique can be applied, for example, to the classical longest increasing subsequence and longest common subsequence problems, extended to labeled DAGs. Finally, we apply this technique to the co-linear chaining problem, which is a generalization of both of these two problems. We also implemented the new co-linear chaining approach. Experiments on splicing graphs show that the new method is efficient also in practice.

A. Tomescu and V. Mäkinen—Shared last author contribution.

1 Introduction

A *path cover* of a DAG $G = (V, E)$ is a set of paths such that every node of G belongs to some path. A *minimum path cover* (MPC) is one having the minimum number of paths. The size of a MPC is also called the *width* of G . Many DAGs commonly used in genome research, such as graphs encoding human mutations [8] and graphs modeling gene transcripts [15], can consist, in the former case, of millions of nodes and, in the latter case, of thousands of nodes. However, they generally have a small width on average; for example, splicing graphs for most genes in human chromosome 2 have width at most 10 [35, Fig. 7]. To the best of our knowledge, among the many MPC algorithms [6, 7, 12, 16, 27, 31], there are only three whose complexities depends on the width of the DAG. Say the width of G is k . The first algorithm runs in time $O(|V||E| + k|V|^2)$ and can be obtained by slightly modifying an algorithm for finding a minimum chain cover in partial orders from [11]. The other two algorithms are due to Chen and Chen: the first one works in time $O(|V|^2 + k\sqrt{k}|V|)$ [6], and the second one works in time $O(\max(\sqrt{|V|}|E|, k\sqrt{k}|V|))$ [7].

In this paper we present an MPC algorithm running in time $O(k|E|\log|V|)$. For example, for $k = o(\sqrt{|V|}/\log|V|)$ and $|E| = O(|V|^{3/2})$, this is better than all previous algorithms. Our algorithm is based on the following standard reduction of a minimum flow problem to a maximum flow problem (see e.g. [2]): (i) find a feasible flow/path cover satisfying all demands, and (ii) solve a maximum flow problem in a graph encoding how much flow can be removed from every edge. Our main insight is to solve step (i) by finding an approximate solution that is greater than the optimal one only by a $O(\log|V|)$ factor. Then, if we solve step (ii) with the Ford-Fulkerson algorithm, the number of iterations can be bounded by $O(k\log|V|)$.

We then proceed to show that some problems (like pattern matching) that admit efficient *sparse dynamic programming* solutions on sequences [10] can be extended to DAGs, so that their complexity increases only by the minimum path cover size k . Extending pattern matching to DAGs has been studied before [3, 24, 28]. For those edit distance -based formulations our approach does not yield an improvement, but on formulations involving a sparse set of matching anchors [10] we can boost the naive solutions of their DAG extensions by exploiting a path cover. Namely, our improvement applies to many cases where a data structure over previously computed solutions is maintained and queried for computing the next value. Our new MPC algorithm enables this, as its complexity is generally of the same form as that of solving the extended problems. Given a path cover, our technique then computes so-called *forward propagation links* indicating how the partial solutions in each path in the cover must be synchronized.

To best illustrate the versatility of the technique itself, in the full version of this paper [19] we show how to compute a longest increasing subsequence (LIS) in a labeled DAG, in time $O(k|E|\log|V|)$. This matches the optimal solution to the classical problem on a single sequence when, e.g., this is modeled as a path (where $k = 1$). In Sect. 4, We also illustrate our technique with the longest

common subsequence (LCS) problem between a labeled DAG $G = (V, E)$ and a sequence S .

Finally, we consider the main problem of this paper—co-linear chaining (CLC)—first introduced in [23]. It has been proposed as a model of the sequence alignment problem that scales to massive inputs, and has been a subject of recent interest (see e.g. [22, 29, 32, 36, 38–40]). In the CLC problem, the input is directly assumed to be a set of N pairs of intervals in the two sequences that match (either exactly or approximately). The CLC alignment solution asks for a subset of these plausible pairs that maximizes the coverage in one of the sequences, and whose elements appear in increasing order in both sequences. The fastest algorithm for this problem runs in the optimal $O(N \log N)$ time [1].

We define a generalization of the CLC problem between a sequence and a labeled DAG. As motivation, we mention the problem of aligning a long sequence, or even an entire chromosome, inside a DAG storing all known mutations of a population with respect to a reference genome (such as the above-mentioned [8], or more specifically a linearized version of it [14]). Here, the N input pairs match intervals in the sequence with paths (also called *anchors*) in the DAG. This problem is not straightforward, as the topological order of the DAG might not follow the reachability order between the anchors. Existing tools for aligning DNA sequences to DAGs (BGREAT [20], vg [25]) rely on anchors but do not explicitly consider solving CLC optimally on the DAG.

The algorithm we propose uses the general framework mentioned above. Since it is more involved, we will develop it in stages. We first give a simple approach to solve a relaxed co-linear chaining problem using $O((|V| + |E|)N)$ time. Then, we introduce the MPC approach that requires $O(k|E| \log |V| + kN \log N)$ time. As above, if the DAG is a labeled path representing a sequence, the running time of our algorithm is reduced to the best current solution for the co-linear chaining problem on sequences, $O(N \log N)$ [1]. In the full version of this paper [19], we use a Burrows-Wheeler technique to efficiently handle a special case that we omitted in this relaxed variant. We remark that one can reduce the LIS and LCS problems to the CLC problem to obtain the same running time bounds as mentioned earlier; these are given for the sake of comprehensiveness.

In the last section we discuss the anchor-finding preprocessing step. We implemented the new MPC-based co-linear chaining algorithm and conducted experiments on splicing graphs to show that the approach is practical, once anchors are given. Some future directions on how to incorporate practical anchors, and how to apply the techniques to transcript prediction, are discussed.

Notation. To simplify notation, for any DAG $G = (V, E)$ we will assume that V is always $\{1, \dots, |V|\}$ and that $1, \dots, |V|$ is a topological order on V (so that for every edge (u, v) we have $u < v$). We will also assume that $|E| \geq |V| - 1$. A *labeled DAG* is a tuple (V, E, ℓ, Σ) where (V, E) is a DAG and $\ell : V \mapsto \Sigma$ assign to the nodes labels from Σ , Σ being an ordered alphabet.

For a node $v \in V$, we denote by $N^-(v)$ the set of in-neighbors of v and by $N^+(v)$ the set of out-neighbors of v . If there is a (possibly empty) path from node u to node v we say that u reaches v . We denote by $R^-(v)$ the set of nodes

that reach v . We denote a set of consecutive integers with interval notation $[i..j]$, meaning $\{i, i + 1, \dots, j\}$. For a pair of intervals $m = ([x..y], [c..d])$, we use $m.x$, $m.y$, $m.c$, and $m.d$ to denote the four respective endpoints. We also consider pairs of the form $m = (P, [c..d])$ where P is a path, and use $m.P$ to access P . The first node of P will be called its *startpoint*, and its last node will be called its *endpoint*. For a set M we may fix an order, to access an element as $M[i]$.

2 The MPC Algorithm

In this section we assume basic familiarity with network flow concepts; see [2] for further details. In the *minimum flow problem*, we are given a directed graph $G = (V, E)$ with a single source and a single sink, with a *demand* $d : E \rightarrow \mathbb{Z}$ for every edge. The task is to find a flow of minimum value (the *value* is the sum of the flow on the edges exiting the source) that satisfies all demands (to be called *feasible*). The standard reduction from the minimum path cover problem to a minimum flow one (see, e.g. [26]) creates a new DAG G^* by replacing each node v with two nodes v^-, v^+ , adds the edge (v^-, v^+) and adds all in-neighbors of v as in-neighbors of v^- , and all out-neighbors of v as out-neighbors of v^+ . Finally, the reduction adds a global source with an out-going edge to every node, and a global sink with an in-coming edge from every node. Edges of type (v^-, v^+) get demand 1, and all other edges get demand 0. The value of the minimum flow equals k , the width of G , and any decomposition of it into source-to-sink paths induces a minimum path cover in G .

Our MPC algorithm is based on the following simple reduction of a minimum flow problem to a maximum flow one (see e.g. [2]): (i) find a feasible flow $f : E \rightarrow \mathbb{Z}$; (ii) transform this into a minimum feasible flow, by finding a maximum flow f' in G in which every $e \in E$ now has capacity $f(e) - d(e)$. The final minimum flow solution is obtained as $f(e) - f'(e)$, for every $e \in E$. Observe that this path cover induces a flow of value $O(k \log |V|)$. Thus, in step (ii) we need to shrink this flow into a flow of value k . If we run the Ford-Fulkerson algorithm, this means that there are $O(k \log |V|)$ successive augmenting paths, each of which can be found in time $O(E)$. This gives a time bound for step (ii) of $O(k|E| \log |V|)$.

We solve step (i) in time $O(k|E| \log |V|)$ by finding a path cover in G^* whose size is larger than k only by a multiplicative factor $O(\log |V|)$. This is based on the classical greedy set cover algorithm, see e.g. [37, Chapter 2]: at each step, select a path covering most of the remaining uncovered nodes.

Such approximation approach has also been applied to other covering problems on graphs, such as a 2-hop cover [9]. More importantly, the approximation-and-refinement approach is similar to the one from [11] for finding the minimum number k of chains to cover a partial order of size n . A *chain* is a set of pairwise comparable elements. The algorithm from [11] runs in time $O(kn^2)$, and it has the same feature as ours: it first finds a set of $O(k \log n)$ chains in the same way as us (longest chains covering most uncovered elements), and then in a second step reduces these to k . However, if we were to apply this algorithm to DAGs, it would run in time $O(|V||E| + k|V|^2)$, which is slower than our algorithm for

small k . This is because it uses the classical reduction given by Fulkerson [12] to a bipartite graph, where each edge of the graph encodes a pair of elements in the relation. Since DAGs are not transitive in general, to use this reduction one needs first to compute the transitive closure of the DAG, in time $O(|V||E|)$.

We now show how to solve step (i) within the claimed running time, by dynamic programming.

Lemma 1. *Let $G = (V, E)$ be a DAG, and let k be the width of G . In time $O(k|E| \log |V|)$, we can compute a path cover P_1, \dots, P_K of G , such that $K = O(k \log |V|)$.*

Proof. The algorithm works by choosing, at each step, a path that covers the most uncovered nodes. For every node $v \in V$, we store $m[v] = 1$, if v is not covered by any path, and $m[v] = 0$ otherwise. We also store $u[v]$ as the largest number of uncovered nodes on a path starting at v . The values $u[\cdot]$ are computed by dynamic programming, by traversing the nodes in inverse topological order and setting $u[v] = m[v] + \max_{w \in N^+(v)} u[w]$. Initially we have $m[v] = 1$ for all v . We then compute $u[v]$ for all v , in time $O(|E|)$. By taking the node v with the maximum $u[v]$, and tracing back along the optimal path starting at v , we obtain our first path in time $O(|E|)$. We then update $m[v] = 0$ for all nodes on this path, and iterate this process until all nodes are covered. This takes overall time $O(K|E|)$, where K is the number of paths found.

This algorithm analysis is identical to the one of the classical greedy set cover algorithm [37, Chapter 2], because the universe to be covered is V and each possible path in G is a possible covering set, which implies that $K = O(k \log |V|)$. \square

Combining Lemma 1 with the above-mentioned application of the Ford-Fulkerson algorithm, we obtain our first result:

Theorem 1. *Given a DAG $G = (V, E)$ of width k , the MPC problem on G can be solved in time $O(k|E| \log |V|)$.*

3 The Dynamic Programming Framework

In this section we give an overview of the main ideas of our approach.

Suppose we have a problem involving DAGs that is solvable, for example by dynamic programming, by traversing the nodes in topological order. Thus, assume also that a partial solution at each node v is obtainable from all (and only) nodes of the DAG that can reach v , plus some other independent objects, such as another sequence. Furthermore, suppose that at each node v we need to query (and maintain) a data structure \mathcal{T} that depends on $R^-(v)$ and such that the answer $\text{Query}(R^-(v))$ at v is decomposable as:

$$\text{Query}(R^-(v)) = \bigoplus_i \text{Query}(R_i^-(v)). \quad (1)$$

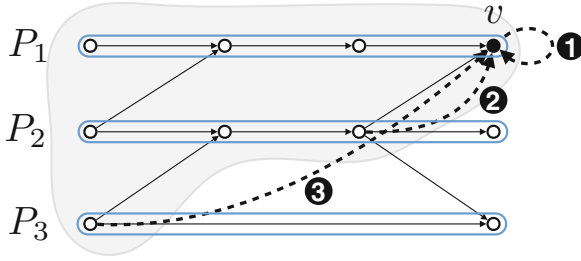


Fig. 1. A path cover P_1, P_2, P_3 of a DAG. The forward links entering v from $last2reach[v, i]$ are shown with dotted black lines, for $i \in \{1, 2, 3\}$. We mark in gray the set $R^-(v)$ of nodes that reach v .

In the above, the sets $R_i^-(v)$ are such that $R^-(v) = \bigcup_i R_i^-(v)$, they are not necessarily disjoint, and \bigoplus is some operation on the queries, such as min or max, that does not assume disjointness. It is understood that after the computation at v , we need to update \mathcal{T} . It is also understood that once we have updated \mathcal{T} at v , we cannot query \mathcal{T} for a node before v in topological order, because it would give an incorrect answer.

The first idea is to decompose the graph into a path cover P_1, \dots, P_K . As such, we decompose the computation only along these paths, in light of (1). We replace a single data structure \mathcal{T} with K data structures $\mathcal{T}_1, \dots, \mathcal{T}_K$, and perform the operation from (1) on the results of the queries to these K data structures.

Our second idea concerns the order in which the nodes on these K paths are processed. Because the answer at v depends on $R^-(v)$, we cannot process the nodes on the K paths (and update the corresponding \mathcal{T}_i 's) in an arbitrary order. As such, for every path i and every node v , we distinguish the *last node* on path i that reaches v (if it exists). We will call this node $last2reach[v, i]$. See Fig. 1 for an example. We note that this insight is the same as in [17], which symmetrically identified the *first* node on a chain i that can be reached from v (a *chain* is a subsequence of a path). The following observation is the first ingredient for using the decomposition (1).

Observation 1. Let P_1, \dots, P_K be a path cover of a DAG G , and let $v \in V(G)$. Let R_i denote the set of nodes of P_i from its beginning until $last2reach[v, i]$ inclusively (or the empty set, if $last2reach[v, i]$ does not exist). Then $R^-(v) = \bigcup_{i=1}^K R_i$.

Proof. It is clear that $\bigcup_{i=1}^K R_i \subseteq R^-(v)$. To show the reverse inclusion, consider a node $u \in R^-(v)$. Since P_1, \dots, P_K is a path cover, then u appears on some P_i . Since u reaches v , then u appears on P_i before $last2reach[v, i]$, or $u = last2reach[v, i]$. Therefore u appears on R_i , as desired. \square

This allows us to identify, for every node u , a set of *forward propagation links* $forward[u]$, where $(v, i) \in forward[u]$ holds for any node v and index i with

$\text{last2reach}[v, i] = u$. These propagation links are the second ingredient in the correctness of the decomposition. Once we have computed the correct value at u , we update the corresponding data structures \mathcal{T}_i for all paths i to which u belongs. We also propagate the query value of \mathcal{T}_i in the decomposition (1) for all nodes v with $(v, i) \in \text{forward}[u]$. This means that when we come to process v , we have already correctly computed all terms in the decomposition (1) and it suffices to apply the operation \oplus to these terms.

The next lemma shows how to compute the values last2reach (and, as a consequence, all forward propagation links), also by dynamic programming.

Lemma 2. *Let $G = (V, E)$ be a DAG, and let P_1, \dots, P_K be a path cover of G . For every $v \in V$ and every $i \in [1..K]$, we can compute $\text{last2reach}[v, i]$ in overall time $O(K|E|)$.*

Proof. For each P_i and every node v on P_i , let $\text{index}[v, i]$ denote the position of v in P_i (say, starting from 1). Our algorithm actually computes $\text{last2reach}[v, i]$ as the index of this node in P_i . Initially, we set $\text{last2reach}[v, i] = -1$ for all v and i . At the end of the algorithm, $\text{last2reach}[v, i] = -1$ will hold precisely for those nodes v that cannot be reached by any node of P_i . We traverse the nodes in topological order. For every $i \in [1..K]$, we do as follows: if v is on P_i , then we set $\text{last2reach}[v, i] = \text{index}[v, i]$. Otherwise, we compute by dynamic programming $\text{last2reach}[v, i]$ as $\max_{u \in N^-(v)} \text{last2reach}[u, i]$. \square

An immediate application of Theorem 1 and of the values $\text{last2reach}[v, i]$ is for solving reachability queries. Another simple application is an extension of the *longest increasing subsequence (LIS)* problem to labeled DAGs. (Both are given in the full version of this paper [19]).

The LIS problem, the LCS problem of Sect. 4, as well as co-linear chaining (CLC) of Sect. 5 make use of the following standard data structure (see e.g. [21, p.20]).

Lemma 3. *The following two operations can be supported with a balanced binary search tree \mathcal{T} in time $O(\log n)$, where n is the number of leaves in the tree.*

- $\text{update}(k, \text{val})$: For the leaf w with $\text{key}(w) = k$, update $\text{value}(w) = \text{val}$.
- $\text{RMaxQ}(l, r)$: Return $\max_{w: l \leq \text{key}(w) \leq r} \text{value}(w)$ (Range Maximum Query).

Moreover, the balanced binary search tree can be built in $O(n)$ time, given the n pairs $(\text{key}, \text{value})$ sorted by component key .

4 The LCS Problem

Consider a labeled DAG $G = (V, E, \ell, \Sigma)$ and a sequence $S \in \Sigma^*$, where Σ is an ordered alphabet. We say that the *longest common subsequence (LCS)* between G and S is a longest subsequence C of any path label in G such that C is also a subsequence of S .

We will modify the LIS algorithm (see the full version of this paper [19]) minimally to find a LCS between a DAG G and a sequence S . The description is self-contained yet, for the interest of page limit, more dense than the LIS algorithm derivation. The purpose is to give an example of the general MPC-framework with fewer technical details than required in the main result of this paper concerning co-linear chaining.

For any $c \in \Sigma$, let $S(c)$ denote set $\{j \mid S[j] = c\}$. For each node v and each $j \in S(\ell(v))$, we aim to store in $\text{LLCS}[v, j]$ the length of the longest common subsequence between $S[1..j]$ and any label of path ending at v , among all subsequences having $\ell(v) = S[j]$ as the last symbol.

Assume we have a path cover of size K and $\text{forward}[u]$ computed for all $u \in V$. Assume also we have mapped Σ to $\{0, 1, 2, \dots, |S|+1\}$ in $O((|V|+|S|) \log |S|)$ time (e.g. by sorting the symbols of S , binary searching labels of V , and then relabeling by ranks, with the exception that, if a node label does not appear in S , it is replaced by $|S|+1$).

Let \mathcal{T}_i be a search tree of Lemma 3 initialized with key-value pairs $(0, 0)$, $(1, -\infty)$, $(2, -\infty)$, \dots , $(|S|, -\infty)$, for each $i \in [1..K]$. The algorithm proceeds in fixed topological ordering on G . At a node u , for every $(v, i) \in \text{forward}[u]$ we now update an array $\text{LLCS}[v, j]$ for all $j \in S(\ell(v))$ as follows: $\text{LLCS}[v, j] = \max(\text{LLCS}[v, j], \mathcal{T}_i.\text{RMaxQ}(0, j-1) + 1)$. The update step of \mathcal{T}_i when the algorithm reaches a node v , for each covering path i containing v , is done as $\mathcal{T}_i.\text{update}(j', \text{LLCS}[v, j'])$ for all j' with $j' < j$ and $j' \in S(\ell(v))$. Initialization is handled by the $(0, 0)$ key-value pair so that any (v, j) with $\ell(v) = S[j]$ can start a new common subsequence.

The final answer to the problem is $\max_{v \in V, j \in S(\ell(v))} \text{LLCS}[v, j]$, with the actual LCS to be found with a standard traceback. The algorithm runs in $O((|V| + |S|) \log |S| + K|M| \log |S|)$ time, where $M = \{(v, j) \mid v \in V, j \in [1..|S|], \ell(v) = S[j]\}$, and assuming a cover of K paths is given. Notice that $|M|$ can be $\Omega(|V||S|)$. With Theorem 1 plugged in, the total running time becomes $O(k|E| \log |V| + (|V| + |S|) \log |S| + k|M| \log |S|)$. Since the queries on the data structures are semi-open, one can use the more efficient data structure from [13] to improve the bound to $O(k|E| \log |V| + (|V| + |S|) \log |S| + k|M| \log \log |S|)$. The following theorem summarizes this result.

Theorem 2. *Let $G = (V, E, \ell, \Sigma)$ be a labeled DAG of width k , and let $S \in \Sigma^*$, where Σ is an ordered alphabet. We can find a longest common subsequence between G and S in time $O(k|E| \log |V| + (|V| + |S|) \log |S| + k|M| \log \log |S|)$.*

When G is a path, the bound improves to $O((|V|+|S|) \log |S| + |M| \log \log |S|)$, which nearly matches the fastest sparse dynamic programming algorithm for the LCS on two sequences [10] (with a difference in $\log \log$ -factor due to a different data structure, which does not work for this order of computation).

5 Co-linear Chaining

We start with a formal definition of the co-linear chaining problem (see Fig. 2 for an illustration), following the notions introduced in [21, Sect. 15.4].

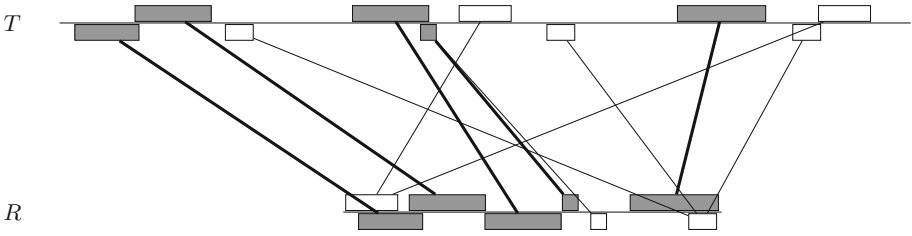


Fig. 2. In the co-linear chaining problem between two sequences T and R , we need to find a subset of pairs of intervals (i.e., anchors) so that (i) the selected intervals in each sequence appear in increasing order; and (ii) the selected intervals cover in R the maximum amount of positions. The figure shows an input for the problem, and highlights in gray an optimal subset of anchors. Figure taken from [21].

Problem 1 (Co-linear chaining (CLC)). Let T and R be two sequences over an alphabet Σ , and let M be a set of N pairs $([x..y], [c..d])$. Find an ordered subset $S = s_1 s_2 \dots s_p$ of pairs from M such that

- $s_{j-1}.y < s_j.y$ and $s_{j-1}.d < s_j.d$, for all $1 \leq j \leq p$, and
- S maximizes the *ordered coverage* of R , defined as

$$\text{coverage}(R, S) = |\{i \in [1..|R|] \mid i \in [s_j.c..s_j.d] \text{ for some } 1 \leq j \leq p\}|.$$

The definition of ordered coverage between two sequences is symmetric, as we can simply exchange the roles of T and R . But when solving the CLC problem between a DAG and a sequence, we must choose whether we want to maximize the ordered coverage on the sequence R or on the DAG G . We will consider the former variant.

First, we define the following *precedence relation*:

Definition 1. Given two paths P_1 and P_2 in a DAG G , we say that P_1 precedes P_2 , and write $P_1 \prec P_2$, if one of the following conditions holds:

- P_1 and P_2 do not share nodes and there is a path in G from the endpoint of P_1 to the startpoint of P_2 , or
- P_1 and P_2 have a suffix-prefix overlap and P_2 is not fully contained in P_1 ; that is, if $P_1 = (a_1, \dots, a_i)$ and $P_2 = (b_1, \dots, b_j)$ then there exists a $k \in \{\max(1, 2 + i - j), \dots, i\}$ such that $a_k = b_1, a_{k+1} = b_2, \dots, a_i = b_{1+i-k}$.

We then extend the formulation of Problem 1 to handle a sequence and a DAG.

Problem 2 (CLC between a sequence and a DAG). Let R be a sequence, let G be a labeled DAG, and let M be a set of N pairs $(P, [c..d])$, where P is a path in G and $c \leq d$ are non-negative integers. Find an ordered subset $S = s_1 s_2 \dots s_p$ of pairs from M such that

- for all $2 \leq j \leq p$, it holds that $s_{j-1}.P \prec s_j.P$ and $s_{j-1}.d < s_j.d$, and
- S maximizes the *ordered coverage* of R , analogously defined as $\text{coverage}(R, S) = |\{i \in [1..|R|] \mid i \in [s_j.c..s_j.d] \text{ for some } 1 \leq j \leq p\}|$.

To illustrate the main technique of this paper, let us for now only seek solutions where paths in consecutive pairs in a solution do not overlap in the DAG. Suffix-prefix overlaps between paths turn out to be challenging; we prove this case in the full version of this paper [19].

Problem 3 (Overlap-limited CLC between a sequence and a DAG). Let R be a sequence, let G be a labeled DAG, and let M be a set of N pairs $(P, [c..d])$, where P is a path in G and $c \leq d$ are non-negative integers (with the interpretation that $\ell(P)$ matches $R[c..d]$). Find an ordered subset $S = s_1 s_2 \dots s_p$ of pairs from M such that

- for all $2 \leq j \leq p$, it holds that there is a non-empty path from the last node of $s_{j-1}.P$ to the first node of $s_j.P$ and $s_{j-1}.d < s_j.d$, and
- S maximizes $\text{coverage}(R, S)$.

First, let us consider a trivial approach to solve Problem 3. Assume we have ordered in $O(|E| + N)$ time the N input pairs as $M[1], M[2], \dots, M[N]$, so that the endpoints of $M[1].P, M[2].P, \dots, M[N].P$ are in topological order, breaking ties arbitrarily. We denote by $C[j]$ the maximum ordered coverage of $R[1..M[j].d]$ using the pair $M[j]$ and any subset of pairs from $\{M[1], M[2], \dots, M[j-1]\}$.

Theorem 3. *Overlap-limited co-linear chaining between a sequence and a labeled DAG $G = (V, E, \ell, \Sigma)$ (Problem 3) on N input pairs can be solved in $O((|V| + |E|)N)$ time.*

Proof. First, we reverse the edges of G . Then we mark the nodes that correspond to the path endpoints for every pair. After this preprocessing we can start computing the maximum ordered coverage for the pairs as follows: for every pair $M[j]$ in topological order of their path endpoints for $j \in \{1, \dots, N\}$ we do a depth-first traversal starting at the startpoint of path $M[j].P$. Note that since the edges are reversed, the depth-first traversal checks only pairs whose paths are predecessors of $M[j].P$.

Whenever we encounter a node that corresponds to the path endpoint of a pair $M[j']$, we first examine whether it fulfills the criterion $M[j'].d < M[j].c$ (call this case (a)). The best ordered coverage using pair $M[j]$ after all such $M[j']$ is then

$$C^a[j] = \max_{j' : M[j'].d < M[j].c} \{C[j'] + (M[j].d - M[j].c + 1)\}, \quad (2)$$

where $C[j']$ is the best ordered coverage when using pairs $M[j']$ last.

If pair $M[j']$ does not fulfill the criterion for case (a), we then check whether $M[j].c \leq M[j'].d \leq M[j].d$ (call this case (b)). The best ordered coverage using pair $M[j]$ after all such $M[j']$ with $M[j'].c < M[j].c$ is then

$$C^b[j] = \max_{j' : M[j].c \leq M[j'].d \leq M[j].d} \{C[j'] + (M[j].d - M[j'].d)\}. \quad (3)$$

Inclusions, i.e. $M[j].c \leq M[j'].c$, can be left computed incorrectly in $C^b[j]$, since there is a better or equally good solution computed in $C^a[j]$ or $C^b[j]$ that does not use them [1].

Finally, we take $C[j] = \max(C^a[j], C^b[j])$. Depth-first traversal takes $O(|V| + |E|)$ time and is executed N times, for $O((|V| + |E|)N)$ total time. \square

However, we can do significantly better than $O((|V| + |E|)N)$ time. In the next sections we will describe how to apply the framework from Sect. 3 here.

5.1 Co-linear Chaining on Sequences Revisited

We now describe the dynamic programming algorithm from [1] for the case of two sequences, as we will then reuse this same algorithm in our MPC approach.

First, sort input pairs in M by the coordinate y into the sequence $M[1], M[2], \dots, M[N]$, so that $M[i].y \leq M[j].y$ holds for all $i < j$. This will ensure that we consider the overlapping ranges in sequence T in the correct order. Then, we fill a table $C[1..N]$ analogous to that of Theorem 3 so that $C[j]$ gives the maximum ordered coverage of $R[1..M[j].d]$ using the pair $M[j]$ and any subset of pairs from $\{M[1], M[2], \dots, M[j-1]\}$. Hence, $\max_j C[j]$ gives the total maximum ordered coverage of R .

Consider Eq. (2) and (3). Now we can use an *invariant technique* to convert these recurrence relations so that we can exploit the range maximum queries of Lemma 3:

$$\begin{aligned} C^a[j] &= (M[j].d - M[j].c + 1) + \max_{j' : M[j'].d < M[j].c} C[j'] \\ &= (M[j].d - M[j].c + 1) + \mathcal{T}.\text{RMaxQ}(0, M[j].c - 1), \\ C^b[j] &= M[j].d + \max_{j' : M[j].c \leq M[j'].d \leq M[j].d} \{C[j'] - M[j'].d\} \\ &= M[j].d + \mathcal{I}.\text{RMaxQ}(M[j].c, M[j].d), \\ C[j] &= \max(C^a[j], C^b[j]). \end{aligned}$$

For these to work correctly, we need to have properly updated the trees \mathcal{T} and \mathcal{I} for all $j' \in [1..j-1]$. That is, we need to call $\mathcal{T}.\text{update}(M[j'].d, C[j'])$ and $\mathcal{I}.\text{update}(M[j'].d, C[j'] - M[j'].d)$ after computing each $C[j']$. The running time is $O(N \log N)$.

Figure 2 illustrates the optimal chain on our schematic example. This chain can be extracted by modifying the algorithm to store traceback pointers.

Theorem 4 ([1, 32]). *Problem 1 on N input pairs can be solved in the optimal $O(N \log N)$ time.*

5.2 Co-linear Chaining on DAGs Using a Minimum Path Cover

Let us now modify the above algorithm to work with DAGs, using the main technique of this paper.

Theorem 5. *Problem 3 on a labeled DAG $G = (V, E, \ell, \Sigma)$ of width k and a set of N input pairs can be solved in time $O(k|E| \log |V| + kN \log N)$ time.*

Proof. Assume we have a path cover of size K and $\mathbf{forward}[u]$ computed for all $u \in V$. For each path $i \in [1..K]$, we create two binary search trees \mathcal{T}_i and \mathcal{I}_i . As a reminder, these trees correspond to coverages for pairs that do not, and do overlap, respectively, on the sequence. Moreover, recall that in Problem 3 we do not consider solutions where consecutive paths in the graph overlap.

As keys, we use $M[j].d$, for every pair $M[j]$, and additionally the key 0. The value of every key is initialized to $-\infty$.

After these preprocessing steps, we process the nodes in topological order, as detailed in Algorithm 1. If node v corresponds to the endpoint of some $M[j].P$, we update the trees \mathcal{T}_i and \mathcal{I}_i for all covering paths i containing node v . Then we follow all forward propagation links $(w, i) \in \mathbf{forward}[v]$ and update $C[j]$ for each path $M[j].P$ starting at w , taking into account all pairs whose path endpoints are in covering path i . Before the main loop visits w , we have processed all forward propagation links to w , and the computation of $C[j]$ has taken all previous pairs into account, as in the naive algorithm, but now indirectly through the K search trees. Exceptions are the pairs overlapping in the graph, which we omit in this problem statement. The forward propagation ensures that the search tree query results are indeed taking only reachable pairs into account. While $C[j]$ is already computed when visiting w , the startpoint of $M[j].P$, the added coverage with the pair is updated to the search trees only when visiting the endpoint.

There are NK forward propagation links, and both search trees are queried in $O(\log N)$ time. All the search trees containing a path endpoint of a pair are updated. Each endpoint can be contained in at most K paths, so this also gives the same bound $2NK$ on the number of updates. With Theorem 1 plugged in, we have $K = k$ and the total running time becomes $O(k|E| \log |V| + kN \log N)$. \square

6 Discussion and Experiments

For applying our solutions to Problem 2 in practice, one first needs to find the alignment anchors. As explained in the problem formulation, alignment anchors are such pairs $(P, [c..d])$ where P is a path in G and $\ell(P)$ matches $R[c..d]$. With sequence inputs, such pairs are usually taken to be *maximal exact matches* (MEMs) and can be retrieved in small space in linear time [4, 5]. It is largely an open problem how to retrieve MEMs between a sequence and a DAG efficiently: The case of length-limited MEMs is studied in [33], based on an extension of [34] with features such as suffix tree functionality. On the practical side, anchor finding has already been incorporated into tools for conducting alignment of a sequence to a DAG [20, 25].

For the purpose of demonstrating the efficiency of our MPC-approach applied to co-linear chaining, we implemented a MEM-finding routine based on simple dynamic programming. We leave it for future work to incorporate a practical procedure (e.g. like those in [20, 25]). We tested the time improvement of

Algorithm 1. Co-linear chaining between a sequence and a DAG using a path cover.

Input: DAG $G = (V, E)$, a path cover P_1, P_2, \dots, P_K of G , and N pairs $M[1], M[2], \dots, M[N]$ of the form $(P, [c..d])$.

Output: The index j giving $\max_j C[j]$.

Use Lemma 2 to find all forward propagation links;

for $i \leftarrow 1$ **to** K **do**

Initialize search trees \mathcal{T}_i and \mathcal{I}_i with keys $M[j].d$, $1 \leq j \leq N$, and with key 0, all keys associated with values $-\infty$;
 $\mathcal{T}_i.\text{update}(0, 0)$;
 $\mathcal{I}_i.\text{update}(0, 0)$;

/ Save to start[i] (respectively, end[i]) the indexes of all pairs whose path starts (respectively, ends) at i. */*

for $j \leftarrow 1$ **to** N **do**

$\text{start}[M[j].P.\text{first}].\text{push}(j)$;
 $\text{end}[M[j].P.\text{last}].\text{push}(j)$;

for $v \in V$ *in topological order* **do**

for $j \in \text{end}[v]$ **do**

/ Update the search trees for every path that covers v, stored in paths[v]. */*

for $i \in \text{paths}[v]$ **do**

$\mathcal{T}_i.\text{update}(M[j].d, C[j])$;
 $\mathcal{I}_i.\text{update}(M[j].d, C[j] - M[j].d)$;

for $(w, i) \in \text{forward}[v]$ **do**

for $j \in \text{start}[w]$ **do**

$C^a[j] \leftarrow (M[j].d - M[j].c + 1) + \mathcal{T}_i.\text{RMaxQ}(0, M[j].c - 1)$;
 $C^b[j] \leftarrow M[j].d + \mathcal{I}_i.\text{RMaxQ}(M[j].c, M[j].d)$;
 $C[j] \leftarrow \max(C[j], C^a[j], C^b[j])$;

return $\text{argmax}_j C[j]$;

our MPC-approach (Theorem 5) over the trivial algorithm (Theorem 3) on the sequence graphs of annotated human genes. Out of all the 62219 genes in the HG38 annotation for all human chromosomes, we singled out 8628 genes such that their sequence graph had at least 5000 nodes. Out of these, we picked 500 genes at random.

The size of the graphs for these 500 genes varied between $|V| = 5023$ and $|V| = 30959$ vertices. Their width, i.e., the number of paths in the MPC, varied between $k = 1$ and $k = 15$. (The number of graphs for each value of k is listed in the column #graphs of the top table of Fig. 3.) The number of anchors, N , for patterns of length 1000 varied between 10^1 and 10^5 . As shown in Fig. 3, with small values of N , our MPC-based co-linear chaining algorithm was twice as fast as the trivial algorithm. When values of N were increased from 10^1 to 10^5 , the difference increased to two orders of magnitude.

k	#graphs	mean $ V $	MPC method	Naive method
1	75	7275	$18 \pm 27\text{ms}$	$5638 \pm 12378\text{ms}$
2	117	8109	$23 \pm 36\text{ms}$	$6355 \pm 17641\text{ms}$
3	93	8306	$27 \pm 41\text{ms}$	$6499 \pm 17940\text{ms}$
4	99	8933	$32 \pm 49\text{ms}$	$6864 \pm 17868\text{ms}$
5	48	9779	$40 \pm 59\text{ms}$	$8053 \pm 18742\text{ms}$
6	32	10265	$45 \pm 65\text{ms}$	$7934 \pm 16659\text{ms}$
7	16	9928	$41 \pm 59\text{ms}$	$6973 \pm 15345\text{ms}$
8	10	11052	$57 \pm 83\text{ms}$	$8731 \pm 17497\text{ms}$
9	4	9538	$52 \pm 77\text{ms}$	$6252 \pm 13906\text{ms}$
10	3	10833	$61 \pm 102\text{ms}$	$7055 \pm 16221\text{ms}$
11	2	11186	$50 \pm 70\text{ms}$	$5932 \pm 10548\text{ms}$
15	1	16848	$154 \pm 194\text{ms}$	$25253 \pm 43873\text{ms}$

N	mean $ V $	MPC method	Naive method
$(10^0..10^1]$	8681	$8 \pm 5\text{ms}$	$15 \pm 8\text{ms}$
$(10^1..10^2]$	8808	$8 \pm 5\text{ms}$	$79 \pm 68\text{ms}$
$(10^2..10^3]$	9732	$10 \pm 7\text{ms}$	$524 \pm 392\text{ms}$
$(10^3..10^4]$	6824	$70 \pm 22\text{ms}$	$15153 \pm 5875\text{ms}$
$(10^4..10^5]$	12235	$153 \pm 66\text{ms}$	$49482 \pm 31900\text{ms}$

Fig. 3. The average running times, and their standard deviation, (in milliseconds) of the two approaches for co-linear chaining between a sequence and a DAG (Problem 2), for all inputs of a certain width k (top), and with N belonging to a certain interval (below). Both approaches are given the same anchors; the time for finding them is not included.

The improved efficiency when compared to the naive approach gives reason to believe a practical sequence-to-DAG aligner can be engineered along the algorithmic foundations given here. Future work includes the incorporation of a practical anchor-finding method, and testing whether the complete scheme improves transcript prediction through improved finding of exon chains [18, 30].

On the theoretical side, it remains open whether the MPC algorithm could benefit from a better initial approximation and/or one that is faster to compute. More generally, it remains open whether the overall bound $O(k|E| \log |V|)$ for the MPC problem can be improved.

Acknowledgements. We thank the anonymous reviewers for comments that improved the presentation of this paper. We thank Gonzalo Navarro for pointing out the connection to pattern matching on hypertexts. This work was funded in part by the Academy of Finland (grant 274977 to AIT and grants 284598 and 309048 to AK and to VM), and by Futurice Oy (to TP).

References

1. Abouelhoda, M.: A chaining algorithm for mapping cdna sequences to multiple genomic sequences. In: Ziviani, N., Baeza-Yates, R. (eds.) SPIRE 2007. LNCS, vol. 4726, pp. 1–13. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75530-2_1
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall Inc, Upper Saddle River (1993)
3. Amir, A., Lewenstein, M., Lewenstein, N.: Pattern matching in hypertext. *J. Algorithms* **35**(1), 82–99 (2000)
4. Belazzougui, D.: Linear time construction of compressed text indices in compact space. In: Proceedings of the Symposium on Theory of Computing STOC 2014, pp. 148–193. ACM (2014)
5. Belazzougui, D., Cunial, F., Kärkkäinen, J., Mäkinen, V.: Versatile succinct representations of the bidirectional Burrows-wheeler transform. In: Bodlaender, H.L., Italiano, G.F. (eds.) ESA 2013. LNCS, vol. 8125, pp. 133–144. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40450-4_12
6. Chen, Y., Chen, Y.: An efficient algorithm for answering graph reachability queries. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 893–902, April 2008
7. Chen, Y., Chen, Y.: On the graph decomposition. In: 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, pp. 777–784, Dec 2014
8. Church, D.M., Schneider, V.A., Steinberg, K.M., Schatz, M.C., Quinlan, A.R., Chin, C.-S., Kitts, P.A., Aken, B., Marth, G.T., Hoffman, M.M., et al.: Extending reference assembly models. *Genome Biol.* **16**(1), 13 (2015)
9. Cohen, E., Halperin, E., Kaplan, H., Zwick, U.: Reachability and distance queries via 2-hop labels. *SIAM J. Comput.* **32**(5), 1338–1355 (2003)
10. Eppstein, D., Galil, Z., Giancarlo, R., Italiano, G.F.: Sparse dynamic programming I: linear cost functions. *J. ACM* **39**(3), 519–545 (1992)
11. Felsner, S., Raghavan, V., Spinrad, J.: Recognition algorithms for orders of small width and graphs of small Dilworth number. *Order* **20**(4), 351–364 (2003)
12. Fulkerson, D.R.: Note on Dilworth’s decomposition theorem for partially ordered sets. *Proc. Am. Math. Soc.* **7**(4), 701–702 (1956)
13. Gabow, H.N., Bentley, J.L., Tarjan, R.E.: Scaling and related techniques for geometry problems. In: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, STOC 1984, pp. 135–143. ACM, New York (1984)
14. Haussler, D., Smuga-Otto, M., Paten, B., Novak, A.M., Nikitin, S., Zueva, M., Miagkov, D.: A flow procedure for the linearization of genome sequence graphs. In: Sahinalp, S.C. (ed.) RECOMB 2017. LNCS, vol. 10229, pp. 34–49. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56970-3_3
15. Heber, S., Alekseyev, M., Sze, S.-H., Tang, H., Pevzner, P.A.: Splicing graphs and EST assembly problem. *Bioinformatics* **18**(Suppl. 1), S181–S188 (2002)
16. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in Bipartite graphs. *SIAM J. Comput.* **2**(4), 225–231 (1973)
17. Jagadish, H.V.: A compression technique to materialize transitive closure. *ACM Trans. Database Syst.* **15**(4), 558–598 (1990)
18. Kuosmanen, A., Norri, T., Mäkinen, V.: Evaluating approaches to find exon chains based on long reads. *Brief. Bioinform.* bbw137 (2017)

19. Kuosmanen, A., Paavilainen, T., Gagie, T., Chikhi, R., Tomescu, A.I., Mäkinen, V.: Using minimum path cover to boost dynamic programming on dags: co-linear chaining extended. CoRR, abs/1705.08754 (2018)
20. Limasset, A., Cazaux, B., Rivals, E., Peterlongo, P.: Read mapping on de Bruijn graphs. BMC Bioinform. **17**(1), 237 (2016)
21. Mäkinen, V., Belazzougui, D., Cunial, F., Tomescu, A.I.: Genome-Scale Algorithm Design. Cambridge University Press, Cambridge (2015)
22. Mäkinen, V., Salmela, L., Ylinen, J.: Normalized N50 assembly metric using gap-restricted co-linear chaining. BMC Bioinform. **13**, 255 (2012)
23. Myers, G., Miller, W.: Chaining multiple-alignment fragments in sub-quadratic time. In: Clarkson, K.L. (ed.) Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, 22–24 January 1995, pp. 38–47. ACM/SIAM, San Francisco (1995)
24. Navarro, G.: Improved approximate pattern matching on hypertext. Theor. Comput. Sci. **237**(1–2), 455–463 (2000)
25. Novak, A.M., Garrison, E., Paten, B.: A graph extension of the positional Burrows-Wheeler transform and its applications. In: Frith, M., Storm Pedersen, C.N. (eds.) WABI 2016. LNCS, vol. 9838, pp. 246–256. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43681-4_20
26. Ntafos, S.C., Hakimi, S.L.: On path cover problems in digraphs and applications to program testing. IEEE Trans. Softw. Eng. **5**(5), 520–529 (1979)
27. Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing, STOC 2013, pp. 765–774. ACM, New York (2013)
28. Park, K., Kim, D.K.: String matching in hypertext. In: Galil, Z., Ukkonen, E. (eds.) CPM 1995. LNCS, vol. 937, pp. 318–329. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60044-2_51
29. Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C.: Salmon provides fast and bias-aware quantification of transcript expression. Nat. Methods **14**(4), 417–419 (2017)
30. Rizzi, R., Tomescu, A.I., Mäkinen, V.: On the complexity of minimum path cover with subpath constraints for multi-assembly. BMC Bioinform. **15**(S–9), S5 (2014)
31. Schnorr, C.-P.: An algorithm for transitive closure with linear expected time. SIAM J. Comput. **7**(2), 127–133 (1978)
32. Shibuya, T., Kurochkin, I.: Match chaining algorithms for cDNA mapping. In: Benson, G., Page, R.D.M. (eds.) WABI 2003. LNCS, vol. 2812, pp. 462–475. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39763-2_33
33. Sirén, J.: Indexing variation graphs. In: 2017 Proceedings of the Nineteenth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 13–27. SIAM (2017)
34. Sirén, J., Välimäki, N., Mäkinen, V.: Indexing graphs for path queries with applications in genome research. IEEE/ACM Trans. Comput. Biol. Bioinf. **11**(2), 375–388 (2014)
35. Tomescu, A.I., Gagie, T., Popa, A., Rizzi, R., Kuosmanen, A., Mäkinen, V.: Explaining a weighted dag with few paths for solving genome-guided multi-assembly. IEEE/ACM Trans. Comput. Biol. Bioinf. **12**(6), 1345–1354 (2015)
36. Uricaru, R., Michotey, C., Chiapello, H., Rivals, E.: YOC, a new strategy for pairwise alignment of collinear genomes. BMC Bioinform. **16**(1), 111 (2015)
37. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2001)

38. Vyverman, M., De Baets, B., Fack, V., Dawyndt, P.: A long fragment aligner called ALFALFA. *BMC Bioinform.* **16**(1), 159 (2015)
39. Vyverman, M., De Smedt, D., Lin, Y.-C., Sterck, L., De Baets, B., Fack, V., Dawyndt, P.: Fast and Accurate cDNA mapping and splice site identification. In: *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms (BIOSTEC 2014)*, pp. 233–238 (2014)
40. Wandelt, S., Leser, U.: RRCA: ultra-fast multiple in-species genome alignments. In: *Dediu, A.-H., Martín-Vide, C., Truthe, B. (eds.) AICoB 2014. LNCS, vol. 8542*, pp. 247–261. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07953-0_20



Modeling Dependence in Evolutionary Inference for Proteins

Gary Larson¹, Jeffrey L. Thorne², and Scott Schmidler³(✉)

¹ Department of Statistical Science, Duke University, Durham, NC, USA

gary.larson@stat.duke.edu

² Departments of Biological Sciences and Statistics, North Carolina State University, Raleigh, NC, USA

thorne@statgen.ncsu.edu

³ Departments of Statistical Science and Computer Science, Duke University, Durham, NC, USA

sschmid@duke.edu

Abstract. Protein structure alignment is a classic problem of computational biology, and is widely used to identify structural and functional similarity and to infer homology among proteins. Previously a statistical model for protein structural evolution has been introduced and shown to significantly improve phylogenetic inferences compared to approaches that utilize only amino acid sequence information. Here we extend this model to account for correlated evolutionary drift among neighboring amino acid positions, resulting in a spatio-temporal model of protein structure evolution. The result is a multivariate diffusion process convolved with a spatial birth-death process, which comes with little additional computational cost or analytical complexity compared to the site-independent model (SIM). We demonstrate that this extended, site-dependent model (SDM) yields a significant reduction of bias in estimated evolutionary distances and helps further improve phylogenetic tree reconstruction.

Keywords: Protein structure · Evolution · Dynamic programming
Phylogeny · Diffusion process

1 Introduction

Protein alignment is an integral part of bioinformatic analyses and is a classic, widely studied problem in computational biology. Existing methods for aligning two or more proteins compare amino acid sequences and/or structures of the proteins, and encompass a variety of algorithms with different strengths and purposes. Such algorithms are a fundamental part of phylogenetic research in particular, where the degree and nature of evolutionary divergence between species is a quantity of interest. Alignment procedures that are widely used in studies of protein evolution are based only on the amino acid sequence and do not incorporate the tertiary (three-dimensional) structure of the proteins. Methods that

do incorporate tertiary structure, such as those mentioned in [1], do not account for the evolution over time of those structures. Recently Challis and Schmidler [2] introduced a stochastic evolutionary model of protein sequence and structure for this purpose; however, their approach, like the vast majority of alignment algorithms, assumes that “sites” (individual amino acid characters, or backbone atom coordinate triples) evolve independently of one other. This assumption is well-known to be violated since amino acid identities and spatial locations are highly dependent due to a combination of physico-chemical constraints and interactions, including bond lengths and excluded volume, hydrophobic and electrostatic attraction and repulsion, hydrogen bonding, and other cooperative effects in forming stable local and global protein structure. Nevertheless, alignment algorithms based on both sequence and structural information typically ignore the correlations induced by these interactions. Ignoring dependence is often justified by the computational intractability of site-dependent models [2,3]. In this paper we demonstrate that in structure-based alignment, as in sequence-based, ignoring site dependence systematically biases evolutionary inference. We present an expanded version of the Challis and Schmidler model which incorporates neighbor dependence without sacrificing computational tractability.

1.1 Motivation

Von Haeseler and Schöniger [4] examined the effect of site dependence on estimates of evolutionary distance between pairs of biological sequences. Using a model of whale mitochondrial DNA evolution whereby the sequence evolves as a collection of independent subsequences, each exhibiting Markovian dependence among its amino acids, the authors demonstrated the tendency to underestimate the true evolutionary distance between two sequences when using a site-independent model. Figure 1a replicates this effect using binary sequences from a nearest-neighbor site-dependent sequence model which does not assume independent subsequences, described in the Appendix A.2. When estimating the divergence time for these sequences under a site-independent version of the same model ($b = 1$ for model in Appendix A.2), the posterior distribution (Fig. 1a) shows significant underestimation of the true value.

Despite a variety of efforts, no site-dependent sequence model has emerged as a widely applicable replacement for commonly used site-independent sequence models [5]. The primary hurdle to doing so is computational - adding realistic dependence generally prohibits the use of efficient alignment algorithms which rely on dynamic programming.

On the other hand, we demonstrate in Sect. 2 that the site-independent *structural* model (SIM) of [2] can be extended to a *site-dependent structural* model (SDM), incorporating site dependence while maintaining the same interpretability and mathematical and computational tractability as the SIM. Thus we can incorporate dependence into the evolutionary structural part of the model in a relatively straightforward way. Using data simulated from the SDM, we find a systematic underestimation effect for structural data due to the independent-site assumption, similar to that observed in sequences (Fig. 1e). The new SDM

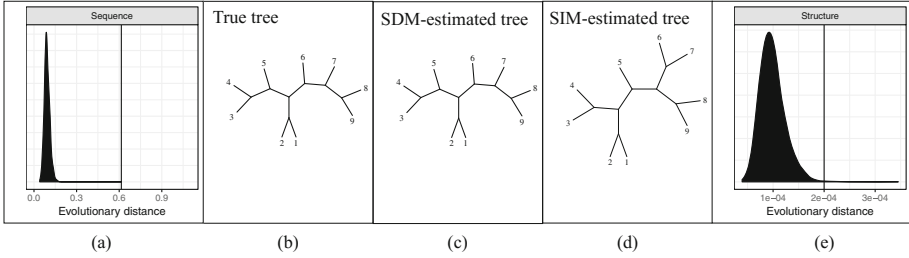


Fig. 1. (a) Posterior distribution of evolutionary distance for sequences simulated under site-dependent model with $b = 2$, $t = 0.6$ (see Sect. A.2), when inference is performed under an assumption of site independence. Significant underestimation is seen relative to truth (vertical line). (b, c, d) This underestimation adversely affects phylogenetic reconstruction, as seen by comparing the true (b) and estimated trees under independent- (d) and dependent-site (c) models. (e) A similar effect is seen for 3D structures, with data simulated under the site-dependent model of Sect. 2.4.

can then be paired with a sequence evolution model to provide a site-dependent expansion of the joint sequence-structure model of Challis and Schmidler [2].

The paper is organized as follows. We briefly review the site-independent structural diffusion model of [2], before describing the general form of a dependent structural diffusion model. Section 2 describes the details of incorporating dependence into the model, with computational tractability being the key constraint on the model’s form. Section 3 describes a reparameterization of the SDM necessary for analyzing the SDM’s effect on phylogenetic inference. Section 4 revisits the motivating example above and compares inferences and phylogenies from the expanded model on a number of real protein examples.

2 A Site-Dependent Structural Diffusion Model

Challis and Schmidler [2] introduced a stochastic model for protein structure evolution, extending a previously developed probabilistic framework for structural alignment of proteins [6, 7] into a model suitable for the study of molecular evolution. This work demonstrated the ability to significantly improve phylogenetic inference when structural information about the proteins is available [2, 3]. We briefly review the original Challis-Schmidler model before introducing our extended model incorporating site dependence. Throughout the paper, these structural models will be referred to as the SIM and SDM respectively.

2.1 Challis-Schmidler Model

Challis and Schmidler [2] model the diffusion of individual C_α backbone positions in space, over time, via an Ornstein-Uhlenbeck (OU) process. Independence is assumed between each site along the backbone as well as between the (x, y, z)

coordinates at each site, leading to the joint structure diffusion being modeled as a product of $3n$ independent univariate OU processes:

$$dC_{ij}^{(t)} = \theta(\zeta_j - C_{ij}^{(t)})dt + \sigma dB \quad (1)$$

where $C_{ij}^{(t)}$ denotes coordinate $j \in \{x, y, z\}$ of α -carbon i at time t . This setup admits tractable stationary and conditional distributions but, as noted by Challis and Schmidler, fails to account for known biophysical interactions which lead to strong observed dependence between sites, such as bond length constraints and the effect of excluded volume in the protein. Although a protein structure's coordinate frame is arbitrarily determined by the experiment, we assume the two structures in our pairwise analyses share a coordinate frame; thus for a pair of structures C^X, C^Y , we assume the coordinate frame of C^X and do not distinguish between C^Y and any rigid body rotation R and translation η thereof. We refer the reader to [2] for a detailed treatment of this issue, and for various other model details omitted here.

2.2 Dependence in a Multivariate Ornstein-Uhlenbeck Process

The independent site model (1) can be written as a multivariate diffusion in the form

$$d\mathcal{C} = -\Theta(\mathcal{C} - \zeta)dt + LdB_t \quad (2)$$

where Θ and $\Sigma = LL'$ are both assumed to be identity matrices. Here the $3n \times 1$ vector $\mathcal{C} = (\mathcal{C}_x, \mathcal{C}_y, \mathcal{C}_z)$ contains the backbone α -carbon coordinates, ζ is the $3n \times 1$ long-term mean vector, and B_t represents $3n$ independent univariate standard Brownian motion terms. Writing the model in this form makes clear that the assumption of site- (and coordinate-) independence can be relaxed by introduction of general Θ and Σ , enabling a more expressive model. For convenience we factor $\Theta = \Sigma_d \otimes \Theta_p$ and $\Sigma = \Sigma_d \otimes \Sigma_p$ as Kronecker products, allowing coordinate dependence (subscript d) and backbone site dependence (subscript p) to be modeled separately.

For purposes of the current paper we set $\Sigma_d = I_3$ allowing the x, y, z dimensions within an individual site to diffuse independently of each other. Observed data suggest that dependence between diffusion in the (x, y, z) dimensions is not strong: Table 1 shows average sample correlations between spatial dimensions for 549 structures comprised of a group of globins and a large group from the manually curated MALIDUP database [8], as well as sample lag-1 autocorrelations (i.e. correlations between consecutive backbone α -carbons) within each spatial dimension. Although some proteins show weak to moderate correlation between spatial dimensions, the averages indicate the correlation is relatively weak compared to the strong autocorrelation along the backbone within a given spatial dimension. Consequently, we focus on incorporating dependence along the backbone rather than among spatial dimensions x, y, z .

Under the SDM then, the joint evolution of the $3n$ scalar coordinates specifying all n backbone positions follows a multivariate OU process governed by

Table 1. Mean sample correlations between dimensions and mean lag-1 autocorrelations along dimensions for 71 globin and 478 MALIDUP protein structures.

	lag-1 autocorrelation			correlation		
	x	y	z	(x,y)	(x,z)	(y,z)
globins	0.95	0.95	0.95	-0.01	0.00	0.01
MALIDUP	0.93	0.93	0.93	0.01	0.02	-0.02

$3n \times 3n$ matrix-valued parameters Θ and Σ . This model introduces site dependence while preserving the analytical tractability of the conditional and limiting distributions of the process, important properties for phylogenetic inference. Under the diffusion process defined by the stochastic differential equation in (2), the joint distribution of $\mathcal{C}^{(t)}$ (the full coordinate set at time t) conditional on $\mathcal{C}^{(s)}$ is multivariate normal:

$$P(\mathcal{C}^{(t)}|\mathcal{C}^{(s)}) \sim N\left(e^{-\Theta\tau}\mathcal{C}^{(s)} + (I - e^{-\Theta\tau})\zeta, \Sigma_\tau\right) \quad (3)$$

with τ denoting the time difference ($t - s$) and with conditional covariance Σ_τ given by

$$vec(\Sigma_\tau) = (\Theta \oplus \Theta)^{-1} \left(I - e^{-(\Theta \oplus \Theta)\tau} \right) vec(\Sigma) \quad (4)$$

where $vec()$ is the linear operator converting a matrix into a column vector. Letting $\tau \rightarrow \infty$ in the conditional mean and covariance gives the stationary distribution

$$P(\mathcal{C}) \sim N(\zeta, \Sigma_\infty) \quad (5)$$

where the stationary covariance Σ_∞ is expressed as

$$vec(\Sigma_\infty) = (\Theta \oplus \Theta)^{-1} vec(\Sigma). \quad (6)$$

Although these closed-form solutions exist for general Σ_p, Θ_p , they are in general not computationally tractable when convolved with the indel process of the evolutionary model from [2] (i.e. the Links model of [9]) because the conditional independence required for dynamic programming is not preserved. To maintain computational tractability in phylogenetic applications, we require forms of Θ_p and Σ_p for which both the conditional and stationary distributions of the multivariate OU exhibit certain conditional independencies, as described in the next section.

2.3 Computational Tractability in Phylogenetic Models

Common uses of evolutionary models, in phylogenetic or homology detection contexts, require the ability to optimize or average over the set of possible alignments. In a Bayesian or maximum likelihood context, the alignment must be

inferred simultaneously with the other parameters. Because of the (exponentially large) size of the alignment space, algorithmic efficiency considerations in these calculations play a key role. In particular, calculating the joint likelihood $p(X, Y)$ of two structures X and Y marginalized over all possible alignments \mathcal{M} is possible in site-independent models by use of dynamic programming (the so-called forward algorithm for pair hidden Markov models (HMMs); see [10]). These algorithms depend on conditional independence properties of the (marginal) likelihood of the backbone coordinates at a single backbone site given all previous backbone sites:

$$P(C_{ij}^X, C_{ij}^Y \mid C_{1j}^X, C_{2j}^X, \dots, C_{(i-1)j}^X, C_{1j}^Y, C_{2j}^Y, \dots, C_{(i-1)j}^Y) = P(C_{ij}^X, C_{ij}^Y) \quad (7)$$

with X and Y denoting ancestor and descendant structures respectively. Models with long-range dependence among sites, including the dependent diffusion model (2) with general $\Theta, \Sigma = LL'$, do not exhibit these conditional independence relationships and therefore prohibit the recursive decomposition which forms the basis of efficient dynamic programming calculations. Since an evolutionary model without efficient alignment algorithms is far too expensive to use in the context of phylogenetic tree inference, we desire a model that incorporates site dependence while still preserving sufficient conditional independence structure to permit use of a forward-type algorithm.

2.4 Constructing a Dependent Structural Diffusion Model

A natural approach to introducing limited neighbor dependence into the diffusion model is to consider the backbone sites' coordinates as a series of nodes with forces acting upon each pair of neighboring sites, for example as in a ball and spring model. Figure 2 shows a general ball and spring model with spring constants k_{ij} . This model corresponds to a probability distribution for the equilibrium positions of the backbones coordinates which has precision matrix $\Sigma^{-1} = (b_{ij})$ where $b_{ij} = b_{ji}$, $b_{ii} = k_{i-1,i} + k_{i,i+1}$ and $b_{ij} = 0$ for $|i - j| > 1$.

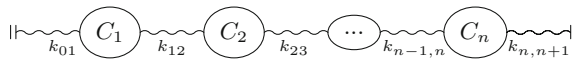


Fig. 2. General ball and spring model for n backbone positions.

The corresponding Gaussian model with neighbor dependence is a spatial first-order auto-regressive process, denoted AR(1). However, setting the spring matrix equal to an AR(1) precision matrix gives a set of equations for the spring constants k_{ij} with no solution. We therefore instead approach the problem of incorporating dependence by starting with a general Θ and Σ and determining what specific forms will correspond to an AR(1) process along the backbone.

We used symbolic algebra software to assist in solving for general matrices Θ_p and symmetric, positive definite Σ_p such that the constraints $\Lambda_\tau(i, j) = \Lambda_\infty(i, j) = 0 \quad \forall i, j : |i - j| > 1$ are satisfied for conditional and stationary

precision matrices A_τ, A_∞ . Solutions to low-dimensional problems allowed us to identify the general form for a single pair of suitable Θ_p, Σ_p . For five backbone positions this nearest-neighbor SDM takes the form:

$$\Theta_p = \theta \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \rho & 1 - \rho^2 & 0 & 0 & 0 \\ \rho^2 & -\rho^3 & 1 & 0 & 0 \\ \rho^3 & -\rho^4 & 0 & 1 & 0 \\ \rho^4 & -\rho^5 & 0 & 0 & 1 \end{pmatrix} \quad \Sigma_p = \sigma^2 \begin{pmatrix} 1 & a\rho & a\rho^2 & a\rho^3 & a\rho^4 \\ a\rho & 1 & \rho & \rho^2 & \rho^3 \\ a\rho^2 & \rho & 1 & \rho & \rho^2 \\ a\rho^3 & \rho^2 & \rho & 1 & \rho \\ a\rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{pmatrix} \quad (8)$$

where $a = (3 - \rho^2)/2$. The conditional and stationary distributions given by (3) and (5) have tri-diagonal precision matrices. Thus dynamic programming is preserved, albeit with some modification to the standard pair HMM recursion formulas required as described in Sect. 2.5.

Similar computer algebra experiments were used to demonstrate that no such solutions exist for any diffusion of the form (2) where $\Sigma_p = I$. With $\Theta = I_3 \otimes \Theta_p$ and $\Sigma = I_3 \otimes \Sigma_p$, (3-6) give the marginal or conditional distributions for matched positions.

2.5 Dynamic Programming

The recursive equations used for the pair hidden Markov model underlying the SIM [10] require several modifications in order to be used with the SDM. These modifications are specific to the form of Θ and $\Sigma = LL'$ chosen for the structural diffusion parameters. The primary reason for the changes is that the backbone coordinate emission probabilities in the SIM are independent of neighboring sites, whereas in the SDM the emission probabilities depend on neighboring sites. The details of the changes required to the dynamic programming algorithm are given in Appendix A.1.

2.6 Bayesian Inference for the Site-Dependent Model

Under the new site-dependent model specified by (2, 8), the joint distribution $p(X, Y | \mathcal{M})$ of backbone coordinates for ancestor X and descendant Y given any alignment \mathcal{M} can be expressed

$$p(X, Y | \mathcal{M}) = \prod_{m \in M} p(X_{[m]}, Y_{[m]} | m, \mathcal{N}_m) \prod_{d \in D} p(X_{[d]}, Y_{[d]} | d, \mathcal{N}_d) \prod_{i \in I} p(X_{[i]}, Y_{[i]} | i, \mathcal{N}_i) \quad (9)$$

where M, D , and I respectively are the sets of matched, deleted, and inserted sites in \mathcal{M} . $X_{[m]}$ denotes the backbone coordinates of the positions of X aligned in $m \in M$, and \mathcal{N}_i is the set of backbone positions neighboring position i . In other words $p(X, Y | \mathcal{M})$ can be expressed in a decomposed form, each factor of which is either the joint density for a contiguous block of matches given its neighbors or the density of an insertion or deletion distribution for a particular site given its neighbors.

Bayesian inference based on this joint distribution (and that including indels) uses priors and sampling techniques detailed in [2] with trivial additions to accommodate priors and sampling for the model's dependence parameter ρ .

3 Joint Sequence-Structure Model for Phylogenetic Inference

Phylogenetic inference involves constructing a phylogenetic tree using estimates of the evolutionary distance between proteins, or equivalently models of the time-dependent evolution. Traditionally this is done using site-independent sequence evolution models parameterized by a matrix Q of relative substitution rates, defining a likelihood over the time τ over which evolution occurs. The joint sequence-structure evolution model introduced by [2] multiplies this likelihood by one derived similarly from the time-dependent structure diffusion process (SIM) given by (1), allowing both structural and sequence differences to inform the estimation of divergence time τ .

3.1 Amino Acid Sequence Model

The sequence portion of our joint sequence and structure model is identical to that used in [2], where the joint likelihood for the two sequences S^X, S^Y and an alignment \mathcal{M} between them is given by

$$\begin{aligned} p(S^X, S^Y, \mathcal{M} | \lambda, \mu, \tau, Q) &= P(S^X, S^Y | \mathcal{M}, \tau, Q) P(\mathcal{M} | \lambda, \mu, \tau) \\ &= P(S_M^Y | S_M^X, \tau, Q) P(S_M^Y | \pi) \times P(S^X | \pi) P(\mathcal{M} | \lambda, \mu, \tau) \end{aligned} \quad (10)$$

where S_M^X, S_M^Y denote the matched (aligned) positions of the amino acid sequences S^X and S^Y , S_M^Y the unmatched positions of S^Y , Q the substitution rate matrix, and π the equilibrium distribution of amino acid labels. The probabilities $P(S_M^Y | S_M^X, \tau, Q)$ are given by a product of independent substitution probabilities at each site via the transition probability matrix $e^{Q\tau}$. $P(S_M^Y | \pi)$ and $P(S^X | \pi)$ are given by the equilibrium distribution π , and we refer the reader to [2] for a discussion of the Links indel model which specifies $P(\mathcal{M} | \lambda, \mu, \tau)$.

3.2 Site-Dependent Random Effect Model

In a sequence evolution model (10), only the product $Q\tau$ is identifiable - one cannot simultaneously estimate absolute rates and τ itself. As a result, it is standard to scale the substitution rate matrix Q to a single expected substitution per unit time [11]. As a result, the time τ is interpreted as the expected number of substitutions per site, which can be estimated from sequences. The structural model exhibits a similar identifiability issue: in pairwise estimation with a structure-only model, with neither rate θ nor time τ fixed, only the structural distance $\theta\tau$ would be identifiable. In the Challis-Schmidler model this was not thought to

be a concern, since when the joint model is used τ becomes determined by the sequence information, making θ identifiable as well.

However this means that disagreement between the structural evolution model and sequence evolution model regarding the divergence time τ will be resolved by compensation in the estimate of θ . Because we do not currently have a computationally tractable site-dependent sequence evolution model, we do not wish the information in the structural SDM to be overridden by the site-independent sequence model, which we know to be susceptible to underestimation. We address this by introducing a distinct sequence time $Q\tau = \tau_q$ and structural time τ_s related by a stochastic model. This differs from the approach of [2, 3], which assumed a common time shared by both structural and sequence components of the likelihood.

The importance of distinguishing these two quantities is highlighted by the plot in Fig. 3, where we estimated divergence time separately using the sequence-only model of (10) and the independent structure-only model (see e.g. [2]) for a set of globins. There is a strong, arguably linear relationship between the structure-only evolutionary distance $\theta\tau$ and the sequence-only evolutionary distance τ , but the relationship between them is clearly noisy. Forcing the two models to share a common parameter ignores the different amounts of information and uncertainty provided about the evolutionary distance by sequence and structural data. The sequence-only and structure-only phylogenetic trees are shown as well, where we see the implications for tree topology.

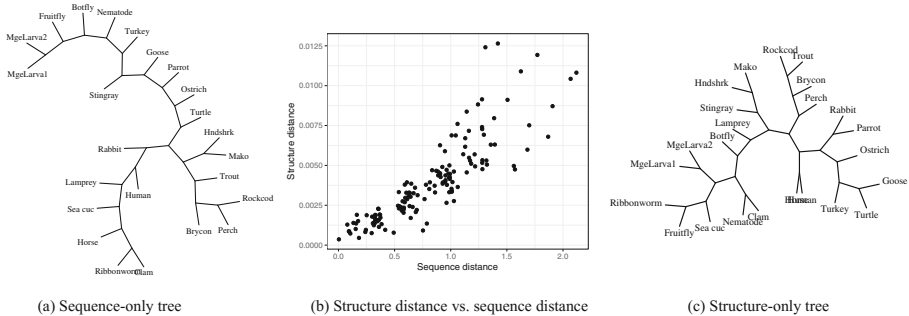


Fig. 3. Pairwise sequence-only distance (τ_q) and structure-only distance ($\theta\tau_s$) estimates from a set of 24 globin proteins under the SIM. The estimates are plotted against each other in panel (b) with the respective phylogenetic tree estimates (via neighbor-joining) in panels (a) and (c). In panel (b), we excluded pairs whose sequence distances could not be reliably estimated due to high sequence divergence.

Instead, we introduce a random effect model defining a stochastic linear relationship between sequence and structure distances:

$$(\theta\tau_s) = \beta\tau_q + \epsilon \quad \text{where } \epsilon \sim N(0, \omega^2). \quad (11)$$

Here τ_s, τ_q are the structural and sequence divergence times respectively. A simple linear regression gives $\hat{\beta} = 0.005$ and an estimate for ω . Under this formulation, the sequence model is now given by

$$\begin{aligned} p(S^X, S^Y, \mathcal{M} | \lambda, \mu, \tau_q, Q) &= P(S^X, S^Y | \mathcal{M}, \tau_q, Q) P(\mathcal{M} | \lambda, \mu, \tau_q) \\ &= P(S_M^Y | S_M^X, \tau_q, Q) P(S_M^Y | \pi) \times P(S^X | \pi) P(\mathcal{M} | \lambda, \mu, \tau_q) \end{aligned} \quad (12)$$

and the PDE governing the structural diffusion is

$$dC = -\Theta(C - \zeta) dt_s + L d\mathbf{B}_{t^{(s)}}. \quad (13)$$

To ensure the structure distance variable τ_s is on a similar scale to τ_q , in each pairwise estimation under this model we fix θ at its posterior mean under the SIM. Hereafter we refer to this joint sequence and structure model with random effect as the SDMre.

4 Results

All inferences were performed on the Duke Computer Cluster (DCC), a heterogeneous network of shared computing nodes; a typical node CPU is an Intel Xeon 2.6 GHz. Average runtimes for the SIM range from 20-60 iterations per second depending primarily on the length of the proteins, while SDM computations are roughly an order of magnitude slower than the SIM. All model parameters were sampled via random-walk Metropolis Hastings, augmented with a library sampling step for rotation parameter R as described in [2].

4.1 Improved Estimation of Evolutionary Distances

We first revisit the example of underestimation in the SIM, shown in Fig. 1(e). The left panel of Fig. 4 shows the posteriors from both the site-independent and site-dependent models. We see again that the SIM underestimates the true evolutionary distance, while the SDM corrects for this.

While this is not surprising on data simulated from the SDM, similar results are observed on real data for which the ‘true’ distance is unknown. The four plots at right in Fig. 4 compare the SIM and SDM posterior distributions for structural distance $\theta\tau$ between two pairs of cysteine proteinases from [3] (top row) and two pairs of globins (human-turtle and human-lamprey, bottom row). In each pairwise estimation, the SIM is significantly underestimating structural distance relative to the SDM. This result is consistently observed across the other pairs of globins and cysteine proteinase pairs from [2, 3] (results omitted for brevity). In each case the SDM posterior is somewhat more diffuse, presumably due to the lower effective sample size in the structural information induced by dependence in the structural model. Although the ‘true’ distances for these pairs cannot be known, these results strongly suggest that including site dependence in the structural model can significantly reduce systematic bias in the estimated evolutionary distances.

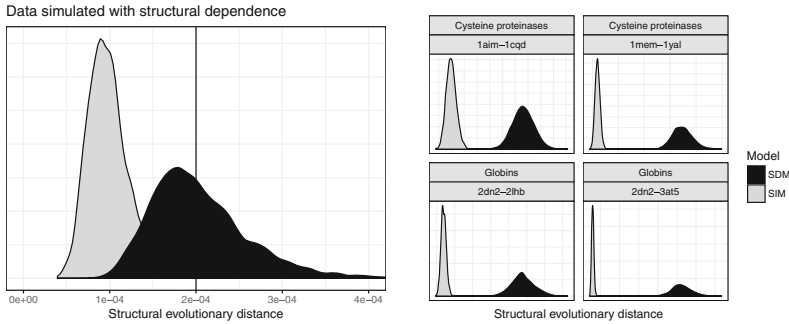


Fig. 4. Estimation of evolutionary distance using SIM (light) and SDM (dark), for (a) simulated data with known true distance, and (b) real data from two cysteine proteinase pairs (b, top row) and two globins (b, bottom row). In all cases the SIM estimate is significantly lower than the SDM estimate, strongly suggesting systematic underestimation under the SIM assumption. Simulation parameters: $\sigma^2 = 1$, $\theta = 0.002$, $t = 0.1$, $\rho = 0.95$.

Non-neighbor dependence: Proteins exhibit significant non-neighbor dependencies due to shared environments and physico-chemical interactions between amino acids that are distant in sequence but proximal in space. Simulations were run using general (non-banded) covariance matrices to simulate structural evolution with long-range correlations, with the SDM then used to estimate evolutionary distance. The results (omitted for brevity) are very similar to the left panel of Fig. 4: the SIM noticeably underestimates the true structural distance while the SDM accurately estimates it. This indicates the robustness of the nearest-neighbor approximation, required for efficient computation, to more general dependency patterns.

4.2 Effect on Phylogeny of Ignoring Structural Dependence in Globin Structures

Errors in estimation of pairwise evolutionary distances have the potential to undermine phylogenetic inference as well. To explore this, we compare phylogenetic trees reconstructed via neighbor-joining for a group of 16 globins using the SIM versus that obtained under the SDMre of Sect. 3. In each case, the respective model was used to estimate the pairwise distances for all pairs of proteins, and the resulting pairwise distance matrix was used to produce a neighbor-joining tree with the PHYLIP and Drawtree software [12]. Differences observed in these trees can be expected to also appear in trees if the SDM were used to replace the SIM component of the fully Bayesian joint sequence-structure tree estimation [3].

The phylogenetic trees estimated using posterior mean evolutionary distances are shown in Fig. 5. The SIM and SDMre trees are very similar, and neither matches the accepted NCBI taxonomy exactly. However, the SDMre tree improves upon the SIM tree in that botfly and fruitfly are now placed together

in a single clade with no other species, as in the NCBI taxonomy. This example demonstrates that phylogeny estimation can be adversely affected by ignoring structural dependence, even for proteins with high structure similarity such as these globins.

The SIM and SDMre models leading to the trees in Fig. 5 differ in two ways: incorporation of dependence in the diffusion, and incorporation of the random-effect relation between the sequence and structure time parameters. For comparison, we also ran the SIM with the random-effect incorporated, but without dependence in the diffusion model. This SIMre does not correctly group botfly and fruitfly, indicating that it is the site dependence which leads to the improved tree topology. For comparison, the sequence-only tree is also shown (for a superset of globins) in panel (a) of Fig. 3; it is highly inaccurate due to many pairs with highly divergent sequences. Without the structural component of the model included, these divergent sequences yield highly uncertain distance estimates which significantly destabilize the tree.

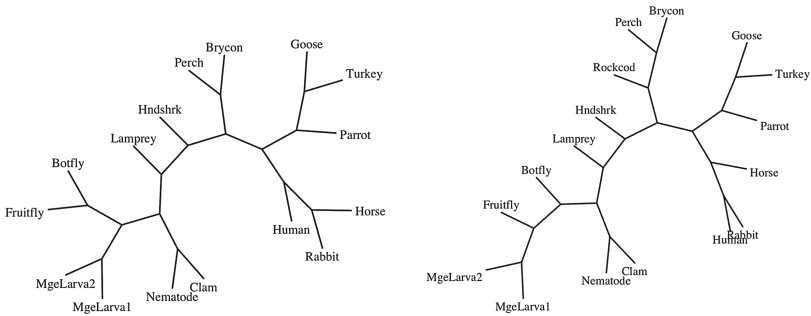


Fig. 5. The SDMre tree (left) improves upon the SIM tree (right) by grouping the botfly and fruitfly in their own clade, matching the accepted NCBI taxonomy.

5 Discussion

The site-dependent structural evolution model described here allows a significant improvement in model realism while retaining the computational tractability necessary for use in phylogenetic inference. As shown, the incorporation of dependence into the model significantly reduces bias in the estimates of evolutionary distance, and can have a resulting stabilizing effect on phylogenetic tree reconstruction. These results suggest a need for continued research on computationally efficient site-dependent *sequence* evolution models, which can be expected to further improve inference in these problems. This is because our current combined sequence-structure model pairs the site-dependent structural model with a site-*independent* sequence model, which likely still retains some downward bias on the estimated evolutionary distance due to the independence assumption in the sequence side of the model.

A natural next step will be to incorporate the site-dependent structural model presented here into the fully Bayesian simultaneous alignment and phylogeny reconstruction model of [3], which currently uses the site-independent structural model. This extension would be straightforward and may improve inference of multiple sequence alignments in addition to improving inference of phylogenetic trees.

Acknowledgments. This work was partially supported by NSF grant DMS-1407622 and NIH grant R01-GM090201 (S.C.S.). Jeffrey L. Thorne was supported by NIH grant GM118508. Gary Larson was partially supported by NSF training grant DMS-1045153 (S.C.S.).

A Appendix

A.1 Modified Dynamic Programming for a Pair HMM with Dependence

In the SDM, the dynamic programming equations' coordinate emission probabilities for each site will now involve preceding positions' coordinates. Because these probabilities are specified by distributions conditional on an alignment, we must know the form of the joint distribution $p(X, Y|\mathcal{M})$ given any alignment \mathcal{M} .

In our model, as in [2], a pair HMM is used to model the distribution of pairwise alignments between two proteins. As described in [10], the use of a pair HMM allows one to calculate the probability of two protein structures marginalized over all possible alignments between the two structures. This is accomplished via dynamic programming by using the well-known forward algorithm to recursively calculate values of $f^k(i, j)$ (i.e., the total probability of all partial alignments through position (i, j) in the ancestor (i) and descendant (j) that end in state $k \in \{\text{Match, Delete, Insert}\}$). The forward equations typically used for this purpose are presented in [10] as:

$$f^M(i, j) = p_{X_i, Y_j} \cdot (a_{MM}f^M(i-1, j-1) + a_{DM}f^D(i-1, j-1) + a_{IM}f^I(i-1, j-1)) \quad (14)$$

$$f^D(i, j) = p_{X_i} \cdot (a_{MD}f^M(i-1, j) + a_{DD}f^D(i-1, j) + a_{ID}f^I(i-1, j)) \quad (15)$$

$$f^I(i, j) = p_{Y_j} \cdot (a_{MI}f^M(i, j-1) + a_{DI}f^D(i, j-1) + a_{II}f^I(i, j-1)) \quad (16)$$

where $p_{X_i, Y_j}, p_{X_i}, p_{Y_j}$ are the three emission probabilities for (respectively): a matched pair X_i, Y_j , a deletion X_i , and an insertion Y_j . Terms of the form a_{JK} give the probability of transition from state J to state K in the pair HMM. The emission probability terms p_{X_i, Y_j}, p_{X_i} and p_{Y_j} involve only the sites denoted and are independent of neighboring sites¹. The SDM emission probabilities are not independent of other sites, so the forward equations must be modified.

¹ For a detailed explanation of the standard forward equation terms we refer the reader to the pair HMM material in [10].

To illustrate the set of changes needed, we focus only on the Match equation (14); analogous changes are required for the other two recursive equations. Equation (14) gives the total probability of all alignments up to position (i, j) which end with a Match at position (i, j) . The three terms on the right hand side arise because a path through the pair HMM could arrive at a Match at (i, j) from one of three previous states in the path: either a Match, Delete, or Insert at $(i - 1, j - 1)$. The term p_{X_i, Y_j} is a single factor on the right hand side, indicating that the Match emission probability at (i, j) is the same regardless of the previous state in the path. In our case, the Match emission probability at (i, j) depends on the previous state in the path. Accordingly, the first step in modifying the equation for our purposes is to define unique emission probabilities that depend on the previous state in the path through the pair HMM. We write the site-dependent version of (14) as

$$\begin{aligned} f^M(i, j) &= (\bar{p}_{X_i, Y_j}^M) \cdot a_{MM} f^M(i - 1, j - 1) \\ &+ (\bar{p}_{X_i, Y_j}^D) \cdot a_{DM} f^D(i - 1, j - 1) \\ &+ (\bar{p}_{X_i, Y_j}^I) \cdot a_{IM} f^I(i - 1, j - 1) \end{aligned} \quad (17)$$

where the superscripts on \bar{p} terms indicate the previous state before the Match at X_i, Y_j . The modified equations for $f^D(i, j)$ and $f^I(i, j)$ are analogous. Any of the emission distributions \bar{p} can be derived by first writing down the joint distribution for the appropriate backbone positions given an alignment (see Sect. 2.6) and then conditioning on that multivariate normal distribution as needed.

When determining the emission distributions, obvious edge cases must be dealt with. In addition, note that the emission distribution for a matched pair given a previous Match (\bar{p}_{X_i, Y_j}^M) depends on where in the alignment the emitted matched pair occurs. In other words, calculation of \bar{p}_{X_i, Y_j}^M should take into account two possibilities: one, that the state prior to the previous Match was also a Match, or two, that it was an Insertion or Deletion. This can be verified by considering the joint distribution for 3 consecutive matched pairs and noting that the distribution of the 2nd matched pair conditional on previous positions is different than the distribution of the 3rd matched pair conditional on previous positions. This characteristic arises due to the specific forms chosen for the OU process' Θ and Σ in our site-dependent model. Thus, the term \bar{p}_{X_i, Y_j}^M in (17) will itself be calculated as a sum over possible states preceding the prior state:

$$\begin{aligned} \bar{p}_{X_i, Y_j}^M &= \bar{p}_{X_i, Y_j}^{(M)_1} [f^D(i - 2, j - 2) \cdot a_{DM} \cdot \bar{p}_{X_{i-1}, Y_{j-1}}^D \\ &+ f^I(i - 2, j - 2) \cdot a_{IM} \cdot \bar{p}_{X_{i-1}, Y_{j-1}}^I] \cdot a_{MM} \\ &+ \bar{p}_{X_i, Y_j}^{(M)_2} [f^M(i - 2, j - 2) \cdot a_{MM} \cdot \bar{p}_{X_{i-1}, Y_{j-1}}^M] \cdot a_{MM}. \end{aligned} \quad (18)$$

The presence of the recursive term $\bar{p}_{X_{i-1}, Y_{j-1}}^M$ in the equation above requires that an additional dynamic programming matrix be tracked. There are no other emission probabilities which depend on more than one previous hidden state of the pair HMM.

Derivation of Emission Probabilities. Suppose \mathcal{M}_p is a known partial alignment of all matches, aligning n positions X_i through X_{i+n-1} to positions Y_j through Y_{j+n-1} with no indels. The joint distribution of these backbone coordinates $p(X_{i,i+n-1}, Y_{j,j+n-1} | \mathcal{M}_p)$ has a block covariance matrix:

$$p(X_{i,i+n-1}, Y_{j,j+n-1} | \mathcal{M}_p) \sim N \left(\mathbf{0}, \begin{pmatrix} \Sigma_{n \times n} & R^T \\ R & \Sigma_{n \times n} \end{pmatrix} \right) \quad (19)$$

where $\Sigma_{n \times n}$ is equal to the stationary OU solution obtained using (8) and R is $n \times n$, equal to:

$$R = \frac{\sigma^2 e^{-\theta\tau}}{2\theta} \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho k & 1 & \rho & \dots & \rho^{n-2} \\ \rho^2 k & \rho & 1 & \dots & \rho^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} k & \rho^{n-2} & \rho^{n-3} & \dots & 1 \end{pmatrix}$$

with $k = \frac{1 - (1 - \rho^2)e^{\theta\rho^2\tau}}{\rho^2}$. The emission probability for an Insertion Y_j or Deletion X_i at a particular site given its previous neighbor has an AR(1) form:

$$p(X_i | X_{i-1}, Y_j) \equiv p(X_i | X_{i-1}) \sim N(\rho X_{i-1}, \sigma^2(1 - \rho^2)) \quad (20)$$

$$p(Y_j | X_i, Y_{j-1}) \equiv p(Y_j | Y_{j-1}) \sim N(\rho Y_{j-1}, \sigma^2(1 - \rho^2)). \quad (21)$$

The joint distribution $p(X, Y | \mathcal{M})$ can be specified by combining these insertion and deletion distributions with the distribution for contiguous matches in (19). Then, the nine dynamic programming emission distributions can be verified using standard techniques for conditioning multivariate normal distributions.

A.2 Dependent Binary Sequence Model

Let σ represent a length n binary sequence. The space of all 2^n possible sequences is $\Omega = \{\sigma_1, \sigma_2, \dots, \sigma_{2^n}\}$. A given sequence σ_i consists of $n-1$ pairs of neighboring labels. To characterize members of Ω , let k_i denote the number of neighbor pairs in σ_i with identical labels (k for “keeps” the same label from one position to the next), and let c_i denote the number of neighbor pairs in σ_i with different labels (c for “changes”). Now define $\lambda_{\sigma_i} := k_i - c_i$. We can refer to λ_σ as a degree of dependence: for sequences with $\lambda_\sigma > 0$, more than half the neighboring label pairs will have the same label and overall the sequence labels will appear non-randomly distributed along the sequence. If $\lambda_{\sigma_i} < 0$, the sequence will look more like a uniform distribution of labels.

To construct a simple model for site-dependent binary sequence evolution, we construct a Markov chain on the state space of binary sequences such that the transitions are site-dependent. We first specify a set of (identical) transition rates $\{a_i\}$ and a corresponding probability jump matrix P having entries P_{ij} . The generator Q for the corresponding Markov chain has entries $Q_{ij} = a_i P_{ij}$. In defining P , we follow the convention that multiple substitutions cannot occur


simultaneously, so that the (i, j) entry of Q and P will be 0 if the configurations σ_i, σ_j differ at more than one position. To induce dependence into such a model, we set $Q_{ij} = b^{\lambda_{\sigma_j} - \lambda_{\sigma_i}} / Z_i$ with $b \geq 1$ an adjustable parameter controlling the strength of neighbor dependence ($b = 1$ represents neighbor independence) and Z_i a normalizing constant for the row such that the off-diagonal row elements sum to 1. Suppose the Markov chain is currently in state i . After an exponential waiting time elapses (given by rate a_i), the Markov chain is more likely to transition to states j having larger $\lambda_{\sigma_j} - \lambda_{\sigma_i}$ than to states j having smaller $\lambda_{\sigma_j} - \lambda_{\sigma_i}$. In other words, in this model a binary sequence is more likely to evolve into a sequence with a more contiguous blocks of identical labels than into a sequence where the sequence labels are uniformly distributed along the sequence length.

References

1. Wang, S., Ma, J., Peng, J., Xu, J.: Protein structure alignment beyond spatial proximity. *Sci. Rep.* **3**, 1448 (2013). <https://doi.org/10.1038/srep01448>
2. Challis, C.J., Schmidler, S.C.: A stochastic evolutionary model for protein structure alignment and phylogeny. *Mol. Biol. Evol.* **29**(11), 3575–3587 (2012). <https://doi.org/10.1093/molbev/mss167>
3. Herman, J.L., Challis, C.J., Novák, A., Hein, J., Schmidler, S.C.: Simultaneous Bayesian estimation of alignment and phylogeny under a joint model of protein sequence and structure. *Mol. Biol. Evol.* **31**(9), 2251–2266 (2014). <https://doi.org/10.1093/molbev/msu184>
4. von Haeseler, A., Schöniger, M.: Evolution of DNA or amino acid sequences with dependent sites. *J. Comput. Biol.* **5**(1), 149–163 (1998). <https://doi.org/10.1089/cmb.1998.5.149>
5. Arenas, M.: Trends in substitution models of molecular evolution. *Front. Genet.* **6**, 319 (2015). <https://doi.org/10.3389/fgene.2015.00319>
6. Schmidler, S.C.: *Bayesian Statistics*, vol. 8. Oxford University Press, New York (2006)
7. Wang, R., Schmidler, S.C.: Bayesian multiple protein structure alignment. In: Sharan, R. (ed.) *RECOMB 2014*. LNCS, vol. 8394, pp. 326–339. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05269-4_27
8. Cheng, H., Kim, B.H., Grishin, N.V.: MALIDUP: a database of manually constructed structure alignments for duplicated domain pairs. *Proteins* **70**(4), 1162–1166 (2008). <https://doi.org/10.1002/prot.21783>
9. Thorne, J.L., Kishino, H., Felsenstein, J.: An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.* **33**(2), 114–124 (1991). <https://doi.org/10.1007/BF02193625>
10. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University, Cambridge (1998). <https://doi.org/10.1110/ps.8.3.695>
11. Kosiol, C., Goldman, N.: Different versions of the Dayhoff rate matrix. *Mol. Biol. Evol.* **22**(2), 193–199 (2005). <https://doi.org/10.1093/molbev/msi005>
12. Felsenstein, J.: *Phylip - phylogeny inference package (version 3.2)*. Cladistics (1989)



Constrained *De Novo* Sequencing of neo-Epitope Peptides Using Tandem Mass Spectrometry

Sujun Li, Alex DeCourcy, and Haixu Tang^(✉) 

School of Informatics, Computing and Engineering,
Indiana University, Bloomington, USA
hatang@indiana.edu

Abstract. Neopeptide peptides are newly formed antigens presented by major histocompatibility complex class I (MHC-I) on cell surfaces. The cells presenting neopeptide peptides are recognized and subsequently killed by cytotoxic T-cells. *Immunoepitidomic* approaches aim to characterize the peptide repertoire (including neopeptide) associated with the MHC-I molecules on the surface of tumor cells using proteomic technologies, providing critical information for designing effective immunotherapy strategies. We developed a novel constrained *de novo* sequencing algorithm to identify neo-epitope peptides from tandem mass spectra acquired in immunoepitidomic analyses. Our method incorporates prior probabilities to putative peptides according to position specific scoring matrices (PSSMs) representing the sequence preferences recognized by MHC-I molecules. We implemented a dynamic programming algorithm to determine the peptide sequences with an optimal posterior matching score for each given MS/MS spectrum. Similar to the *de novo* peptide sequencing, the dynamic programming algorithm allows an efficient searching in the entire peptide sequence space. On an LC-MS/MS dataset, we demonstrated the performance of our algorithm in detecting the neopeptide peptides bound by the HLA-C*0501 molecules that were superior to database search approaches and existing general purpose *de novo* peptide sequencing algorithms.

Keywords: *De novo* · neo-epitope · Mass spectrometry · Proteomics

1 Introduction

The peptide epitopes presented by major histocompatibility complex class I (MHC-I) molecules on cell surfaces display a representative image of the collection of (endogenously synthesized or exogenous) proteins in the cell, allowing immune cells (e.g., the *CD8*⁺ cytotoxic T-cells) to monitor the biological activities occurring inside the cell, a process known as the *immune surveillance* [2, 7, 28]. A typical process of the peptide processing and presentation involves three steps: (1) the cytosolic proteins are first degraded into peptides by the

proteasome; (2) the resulting peptides are loaded onto MHC-I molecules; and (3) the MHC-I/peptide complex is transported into the plasma membrane of the cell via endoplasmic reticulum (ER), while the extracellular domain of MHC-I, where the epitope peptide binds, is exported outside the membrane. In normal cells, the peptides presented by MHC-I will not induce immune responses. However, when abnormal processes (e.g., viral infection or tumorigenesis) occur inside cells, a fraction of MHC-I molecules may present peptides from foreign or novel proteins (e.g., due to somatic mutations in tumor cells), often referred to as the *neopeptide peptides* or *neoantigens*. Consequently, the cells presenting such peptides will likely to be recognized and subsequently killed by cytotoxic T-cells.

It is now well known that, during tumor development, maintenance and progression, tumor cells accumulate thousands of somatic mutations, many of these occurring in protein-coding regions of tumor genes [6,22,29]. Among them, missense or frameshift mutations have the potential to generate neopeptide peptides, which can be used as biomarkers for characterizing the states and subtypes of cancer, or can be selected as potential therapeutic cancer vaccines to induce robust and tumor-specific responses [7,30]. Furthermore, neopeptide peptides were recently demonstrated as potential targets in cancer immunotherapies such as adoptive T-cell therapy [39].

In the past decade, clinical evidence has been accumulated on tumor-specific immune activities, leading to the implementation of successful strategies of cancer immunotherapy [9]. Because of the strong implications of neopeptide peptides in the design of effective cancer immunotherapy, different genomic and proteomic methods have been developed to identify neopeptide peptides presented by tumor cells from cancer patients. The genomic approaches start from exon and transcriptome sequencing of normal and tumor tissues in attempt to identify proteins over- or under-expressed tumor issues, as well as missense or frameshift mutations in tumor proteins [20,25], and then use computational methods [1,13,40] to predict neopeptide candidate from these tumor proteins based on the *immunogenicity* of peptides, i.e., the likelihood of peptides being presented by MHC-I molecules in tumor cells and furthermore likely to provoke an immune response. Notably, the genomic approaches may not report accurate neopeptide peptides due to various limitations of the methods. First, some very low abundant proteins that may not be identified using transcriptome sequencing are often presented by the MHC-I molecules, and can provoke robust immune responses. Second, current immunogenicity prediction algorithms cannot yet accurately model the process of antigenic peptide processing and presentation by MHC-I, and thus may report many false positives and false negatives of neopeptide peptides. Most importantly, as multiple MHC-I molecules are encoded by the highly polymorphic human leukocyte antigen (HLA) genes (including three major types of HLA-I, HLA-II and HLA-III) in an individual patient, the *peptide immunogenicity* is indeed a private measure specific to this cancer patient, and thus cannot

be modeled without sufficient neoepitope peptides already identified from the patient's own sample [10].

In contrast, the *immunopeptidomic* approaches aim to directly analyze the peptide repertoire bound by the MHC-I molecules on the surface of tumor cells using proteomic technologies, and thus can overcome the limitations of genomic approaches. Because of its high throughput and sensitivity, liquid chromatography coupled tandem mass spectrometry (LC-MS/MS) has been routinely used in proteomics in an attempt to identify and quantify proteins in complex protein mixtures, and also becomes the technology of choice for the identification of neoepitope peptides eluted from MHC molecules [5]. From the MS/MS spectra acquired in an immunopeptidomic experiment, potential neoepitope peptides are identified often using a database search engines designed for peptide identification in proteomics (e.g. Sequest [12], Mascot [8] or MSGF+ [19]). However, the neoepitope peptides have some distinct features comparing to the peptides from general proteomic analysis. On one hand, neoepitope peptides bound to different classes of MHC-I molecules have relatively fixed length; for example, human HLA class I (HLA-I) recognizes peptides 8 to 12 amino acid residues in length [4].

On the other hand, unlike the peptides in proteomic experiments typically from tryptic digestion at specific basic amino acid residues, neoepitope peptides can be cleaved by proteasome at any arbitrary position in the target proteins. As a result, when MS/MS spectra from an immunopeptidomic study is searched against a target protein database (e.g. consisting of all human proteins), all non-tryptic peptides of the lengths within a range (8–12 residues) are considered; in the human protein database, there are $\approx 10^7 - 10^8$ such peptides, much greater than the number of tryptic peptides ($\approx 10^6$). Furthermore, a recent study demonstrated that a surprisingly large fraction (about a third) of neoepitope peptides are generated by *proteasome-catalyzed peptide splicing* (PCPS) that cuts and pastes peptide sequences from different proteins [24]. If all concatenate peptides (with two subpeptides from the same or different proteins) are considered in the database search, the number of target peptides increases to $\approx 10^{15}$, close to the total number of peptides 8–12 residues in length. Which poses great challenges to database search not only on the running time but also on potential false positives in peptide identification. Finally, strong sequence patterns are present in neoepitope peptides, largely because of the preferences in the binding affinity

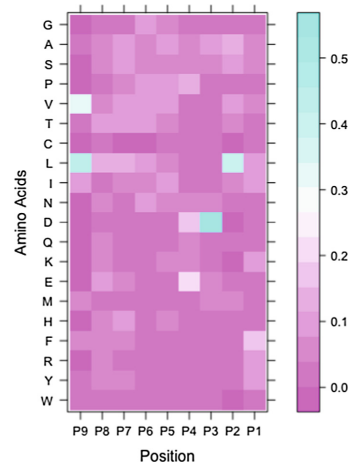


Fig. 1. An example of positional specific scoring matrix (PSSM) (shown as a frequency heatmap) derived from neoepitope peptides of 9 amino acid residues bound to HLA-C*0501. The third position is dominated by Asp while at the ninth position, Leu and Val are preferred.

and specific structures of MHC-I molecules. The sequence pattern in neopeptide peptides recognized by a specific class of MHC-I molecule can be represented by a positional specific scoring matrices (PSSMs; see Fig. 1 as an example for HLA-C) [17], or more complex machine learning models for predicting peptide immunogenicity [1]. However, these sequence information are not used by current approaches for neopeptide peptide identification in proteomic experiments.

De novo peptide sequencing algorithms (such as Peaks [27], pepNovo [14], pepHMM [37] and most recently, uniNovo [16], Novor [26] and DeepNovo [32, 33]) represent a different approach to peptide identification in proteomics, that attempt to reconstruct the peptide sequence directly from an MS/MS spectrum. Comparing to database search algorithms, *de novo* sequencing algorithms explore the entire space of peptides, but are often more efficient because of the employment of a dynamic programming algorithm. From a Bayesian perspective, the database search approach can be viewed as a special case of *de novo* peptide sequencing, which assumes that only the proteins in the database can be present in the sample, and thus the peptides from these proteins have the prior probabilities of 1 while the other peptides have the prior probabilities of 0 [23]. Previous studies have showed that although the top peptide reported by the *de novo* sequencing algorithm for an MS/MS spectrum was sometimes incorrect, the correct one was usually the peptide in the database that received the highest score in *de novo* sequencing [14, 27], indicating that the incorporation of the protein database as prior knowledge significantly improves peptide identification.

In this paper, we present a novel constrained *de novo* sequencing algorithm for neopeptide peptide identification. The method can be viewed as a hybrid approach of the *de novo* sequencing and the database searching algorithms: it explores the entire space of peptide sequences 9–12 residues in length, but assigns a different prior probability to each putative peptide according to MHC-I specific PSSMs, such that the peptide with a motif with high immunogenicity incorporates a high prior probability into the posterior probability score of the peptide-spectrum matches (PSMs). Utilizing the sequential property of the PSSMs, we extended the dynamic programming (DP) algorithm for *de novo* peptide sequencing to determine the peptide sequences with the optimal posterior matching scores for each given MS/MS spectrum. Notably, similar to *de novo* peptide sequencing algorithms, the dynamic programming algorithm allows an efficient searching in the entire peptide sequence space, which, as shown above, is comparable to the size of the database consisting of all putative neopeptide peptides (including the concatenate peptides) derived from human proteins. We tested our algorithm in a LC-MS/MS dataset for detecting the neopeptide peptides bound by the HLA-C*0501 molecules [18]. Our method could detect about 19,017 neopeptide peptides of lengths between 9 to 12 residues with estimated false discovery rate below 1%. In contrast, the database search approach (using MSGF+ against the human protein database) identified about 4,415 PSMs (1,804 unique peptides), in which 2,104 PSMs (764 unique peptides) have the length between

9 to 12 residues as putative neopeptide peptides. Out of the 2,104 PSMs, 1,269 were also identified by our method. A majority (791 out of 1,269) of the PSMs were exact matches, while most (360 out of 478) remaining PSMs contain only a swap of consecutive residues in peptide sequences. Finally, we tested a conventional *de novo* sequencing algorithm uniNovo [16] on the same dataset. It reported sequence tags on 1,863 MS/MS spectra, but with low sequence coverage (on average three amino acid residues per peptide), and thus cannot be used in neopeptide peptide sequencing. These results imply that the constrained *de novo* sequencing algorithm benefit from the prior probabilities (provided by the PSSMs) to distinguish the most likely neopeptide peptides from other peptides sharing similar sequences.

2 Method

Constrained *de novo* Peptide Sequencing. Given an MS/MS spectrum M , the *constrained de novo* peptide sequencing problem is to find the peptide sequence T within a range of length ($l_{min} \leq |T| \leq l_{max}$) that maximizes a posterior matching score S :

$$Score(M, T) = P(T) \cdot P(M|T) \quad (1)$$

where $P(T)$ represents the prior probability of the peptide T , and $P(M|T)$ represents the matching probability, i.e., the probability of observing the MS/MS spectra from the peptide T . For peptides with a fixed length l , their prior probabilities are defined by a PSSM p_{ij} ($\sum_i p_{ij} = 1$) for residue i at the position j ($j = 1, 2, \dots, l$) in the peptide; thus, for the peptide $T = t_1 t_2 \dots t_l$, $P(T) = \prod_{j=1}^l p_{t_j j}$. The matching probability $P(M|T)$ is modeled by the independent fragmentation at each peptide bond: $P(M|T) = \prod_{j=1}^l 1P(f_{M,j})$, where $P(f_{M,j})$ stands for the probability of observing $f_{M,j}$, the occurrence pattern of the set of fragment ions, including the *b*-ion, *y*-ion and the neutral loss ions, derived from the fragmentation between the precursor ($t_1 t_2 \dots t_j$) and the suffix ($t_{j+1} t_{j+2} \dots t_l$) peptide in M . Notably, $f_{M,j}$ is dependent only on m_j , the *j*-th prefix mass of the prefix peptide $t_1 t_2 \dots t_j$, but is not dependent on the peptide sequences. Therefore,

$$Score(M, T) = \prod_{j=1}^l [p_{t_j j} P(F(m_j))] \quad (2)$$

where $P(F(m_j))$ represents probability of observing the set of fragment ion $F(m_j)$ associated with the prefix mass m_j in M . These probabilities can be learned from a training set of identified MS/MS spectra [14], in which the peaks are assigned. Alternatively, as adopted here, $P(F(m_j))$ is assigned empirically based on the logarithm transformed ion intensities of the matched *b*- or *y*-ions

(within a mass tolerance). Let $S(j, m)$ be the maximum posterior matching score between an MS/MS spectrum and any peptide of length j with a total mass of m , which can be computed by using a dynamic programming algorithm,

$$S(j, m) = \max_{k \in A} [S(j-1, m-k) \cdot [p_{j,k} \cdot P(F(m))]] \quad (3)$$

where k is an amino acid in the alphabet A . Note that the multiplication of probabilities in Eq. 3 can be transformed into the summation of the logarithms of probabilities. Finally, the optimal potential matching score of a peptide with a fixed length l , implicated as the number of columns in the PSSM, matching a given spectrum M , is $S(M; l, m_{pr})$, in which m_{pr} is the precursor mass of M . The algorithm can be applied to each putative peptide length between l_{min} and l_{max} with a corresponding PSSM, and the peptides will be reported in the order of their posterior matching scores. The dynamic programming algorithm is executed in $O(l \cdot m_{pr})$ time using $O(l \cdot m_{pr})$ space (where the fragment ion masses are binned according to the mass resolution), but can be further accelerated by heuristics as described below. Note that the prefix mass scoring has been previously proposed as a useful tool for *de novo* peptide sequencing [14], database searching [19] and spectrum alignment to identify mutations and post-translation modifications (PTMs) [31]. The dynamic programming algorithm presented here can be view as matching a predefined PSSM against a vector of prefix mass scores (probabilities) in order to find the optimal matches between a peptide and a subset of prefix masses.

Accelerating the Dynamic Programming Algorithm. For an input MS/MS spectrum of the precursor mass m_{pr} and a PSSM with a specific neopeptide peptide length l , the above algorithm explores all potential prefix masses between 0 and m_{pr} for each prefix peptide of the length from 0 to l . However, there are only a limited number of prefix masses corresponding to prefix peptides of a fixed length, indicating that the matrix of $S(j, m)$ computed in Eq. 3 has many zeroes, especially when for small j . To compute only the non-zero elements in $S(j, m)$, we exploited a branch-and-bound approach to explore the peptide space, while retaining only the best scored sub-peptide among those with the same prefix mass.

The sequencing algorithm maintains a pool of putative prefix peptides, each associated with a posterior matching score. The pool starts with N ($N = |A| = 20$ representing the number of amino acid masses) prefix peptides of length 1 (Fig. 2) with posterior matching scores of $S(1, m(k)) = p_{1k} \cdot P(F(m(k)))$ (where $m(k)$ is the mass of the amino acid k). At each following iteration j , for $j = 2, \dots, l$, every prefix peptide in the pool generates N new prefix peptides, one for every amino acid, by appending a new amino acid to the end of each existing peptide (of length $j - 1$) in the pool.

After appending an amino acid k to an existing prefix peptide with mass m' , the mass of the resulting prefix peptide (i.e., the prefix mass m) is used to compute $P(F(m))$, and then the posterior matching score of the new prefix peptide is computed by $S(j, m) = S(j - 1, m') \cdot p_{jk} \cdot P(F(m))$, where $S(j - 1, m')$ is the posterior matching score associated with the existing prefix peptide of length $j - 1$. At each step, the precursor mass m should match at least one of b- and y-ions; otherwise, the precursor peptide is labeled with one *miscleavage*, which is tracked on each iteration of an algorithm: if a prefix peptide contains too many miscleavages, it is eliminated from further extension. Once the posterior matching score of a prefix peptide is obtained, it will be compared with other peptides in the pool with the same prefix mass, and the k (default = 5) best scoring peptides are retained. After each step, at most $N \times m_{pr}$ prefix peptides are retained in the pool. The algorithm is illustrated in Fig. 2. We note that, although the worst-case running time of the *de novo* sequencing algorithm is still $O(l \cdot m_{pr})$ for each spectrum, in practice, it runs much faster as many un-realistic prefix masses were not evaluated, especially for small l .

In the final step (with prefix peptides of the expected length l), all peptides with masses matching the precursor mass are re-assessed by using a global scoring scheme (see below), and are reported in the order of their global scores. Note that for each input MS/MS spectrum, the constrained *de novo* algorithm was conducted four times, with an input PSSM for peptides of length 9, 10, 11 and 12, respectively.

Pre-processing of MS/MS Spectra. Prior to constrained *de novo* sequencing algorithm, several pre-processing steps were conducted on the MS/MS spectra, including: (1) peaks with an intensity of 0 were removed; (2) the precursor peak was removed; (3) any converted mass greater than precursor mass was removed; (4) Isotopic masses of precursor masses were removed; (5) the intensities of all peaks were logarithm-transformed.

Construction of PSSMs. Peptides of length 9–12 were extracted from the IEDB [35] database <http://www.iedb.org/>, and separated by length. A total of 892 peptides of length 9, 191 peptides of length 10, 110 peptides of length 11,

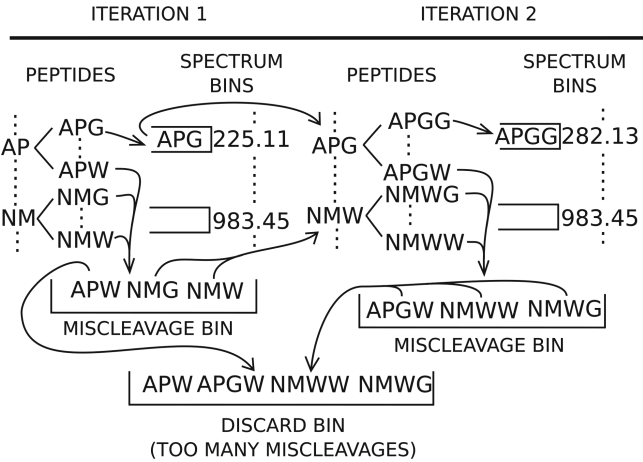


Fig. 2. A schematic illustration of the exploration of the peptide sequence space in the constrained *de novo* algorithm (see text for details).

and two peptides of length 12 were considered. Four PSSMs were created, one for each peptide length, in which the amino acid frequency in every position in the PSSM was computed based on these peptide sequences and the pseudo-count of 1 was incorporated to ensure there were no frequencies of 0.

Re-Assessment of Peptide-Spectrum Matches (PSMs) by Global Scoring. The global score of a PSM is a probability measure, based on a combination of the prior probability based on the input PSSM, and how well it's theoretical fragmentation of the peptide matches to the experimental spectrum. It is calculated using Eq. (1), where $P(T)$ is the probability of the peptide given the PSSM, normalized to the length of the peptide, and $P(M|T)$ is the probability of observing MS/MS M from peptide T based off of the theoretical fragmentation of T . $P(M|T)$ is calculated by

$$\text{Score}(A, E, W) = 1 - \sum_{i=1}^k \frac{a_i \cdot e_i}{W} \quad (4)$$

where e_i is a normalized intensity of the experimental spectrum E , a_i is the mass accuracy (in ppm) between experimental mass i and theoretical fragmentation mass i (or W if there is no matching mass between the two), from the mass accuracy vector A , W is the lowest allowable mass accuracy between an experimental and theoretical mass, and k is the number of peaks in the experimental spectrum M .

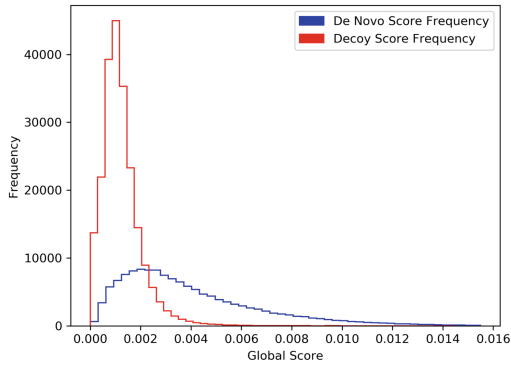


Fig. 3. Score distributions of PSMs reported by the constrained *de novo* sequencing algorithm and the decoy PSMs from the reverse peptides.

False Discovery Rate Estimation. After the global scores were computed for all PSMs, it was necessary to determine a score threshold to validate whether a peptide match was reliably identified from an MS/MS spectrum by our constrained *de novo* sequencing algorithm. Note that it is possible for multiple similar peptide sequences to score high enough to indicate that any of them could be

the correctly identified neoepitope peptide producing the corresponding MS/MS spectrum. In this case, the *de novo* sequencing algorithm reports all of them. As shown in the results section, in practice, usually only a few peptides(2) are reported for each spectrum.

To obtain an appropriate score threshold, we adopted similar strategy to the target-decoy search in database searching [11] to estimate the false discovery rate (FDR) of PSMs. We generated a decoy peptide database consisting of about 40 million randomly selected and reversed peptides with lengths of 9–12 residue from the proteins in the Uniprot database. Additionally, a second database was created for the reversed peptides found by the constrained *de novo* sequencing algorithm. For each spectrum in our analysis, up to 10 peptides matching the spectrum precursor mass within the mass resolution (35 ppm) were selected from both databases as decoys. The top scoring peptides among these decoy peptides were used to form the decoy PSMs, whose global scores were computed. The score distributions are depicted in Fig. 3, containing the scores from both decoy PSMs and the PSMs reported by the constrained *de novo* sequencing algorithm. We then used the following formula to estimate the FDR at a certain score threshold t : $FDR_t = N_{decoy}/N_{cons}$, where N_{decoy} and N_{cons} represent the numbers of decoy and positive (from the sequencing algorithm) PSMs with global scores above t , respectively. We then estimated that PSMs with higher than 0.0058 have FDR lower than 1%.

Datasets. The dataset was obtained from ProteomeXChange [36] (accession number: PXD006455). The experiments were conducted on two common HLA-C: HLA-C*05:01 and HLA-C*07:02. These HLA class I molecules were isolated from the cell surface of C*05 and C*07 transfected 721.221 cells, and sequenced bound peptides by mass spectrometry. As observed in the original article [18], HLA-C*05:01 has higher expression level and more diversified binding peptides. In our testing, we chose the binding peptides of HLA-C*05:01 (with length between 9 to 12 residues) to demonstrate the performance of our method. In total, there are 339,513 spectra acquired in a total 25 fractions of LC-MS/MS analysis using the Q Exactive HF-X MS (Thermo Fisher Scientific) [36].

Database Searching. We used MSGF+ [19] here as the database searching engine. The parameters for the MSGF+ are set as following to match the experimental conditions of the LC-MS/MS analyses: (1) instrument type: high-resolution LTQ; (2) the enzyme type: unspecific cleavage; (3) precursor mass tolerance: 35 ppm; (4) isotope error range: $-1, 2$; (5) modifications: oxidation as variable and carboamidomethyl as fixed; (6) maximum charge is 7 and minimum charge is 1. The FDR is estimated by using a target-decoy search approach (TDA) [11].

3 Results

Constrained *de novo* Sequencing. We implemented the constrained *de novo* sequencing algorithm in C. It spends a total of 8,910 min on a Linux computer

(Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60 GHz) as single thread to process 339,513 input MS/MS spectra in the HLC-C peptidomic dataset, i.e., about 1.6 s per MS/MS spectrum. Among the entire set of spectra, the sequencing algorithm reported one or more peptide sequences for 136,249 (40.14%) spectra, resulting a total of 2,775,977 peptide-spectrum matches (PSMs), i.e., 20 PSMs (peptides) per spectra. Among them, 81,888 PSMs over 28,759 spectra (i.e., 2.85 PSMs per spectra) received a global matching score above 0.0058 (corresponding to about 1% FDR; see Methods), corresponding to 57,449 unique peptides, are retained for further analysis.

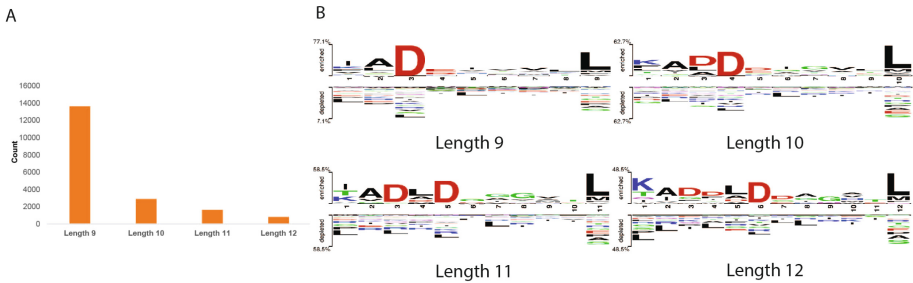


Fig. 4. The length distributions of the top-ranked peptides reported by the constrained *de novo* sequencing algorithm (A); and the sequence logos representing the position specific frequency pattern among the top-ranked peptides with different lengths (B).

The top-ranked peptides of the 28,759 spectra corresponds to 19,017 unique peptides. The length distribution of these peptides is illustrated in Fig. 4. A majority (13,648, 71.76%) of them are 9 residues in length, which is consistent with previous observations [18] and the IEDB database [35], in which 892 out of 1,195 (74.64%) HLA-C*0501 bounded peptides are 9 residues in length. Figure 4B shows the sequence logo [34] generated by using the identified peptides by the *de novo* sequencing method. Specifically, 13,648 peptides have 9 residues, 2,904 have 10 residues, 1,647 have 11 residues, and 818 have 12 residues. Those sequences were used to generate the sequence logos in Fig. 4. For peptides of length 9, the sequence logo showed that the positions of P2, P3 and P9 have strong amino acid preferences: P2 is enriched by Ala, P9 is enriched by Leu/Ile, and P3 is dominated by Asp. For peptides of other lengths, Asp is predominant at multiple positions, especially in the peptides N-termini, while Leu/Ile are predominant in peptides C-termini.

If all the sequences are retained as long as the global matching score is above the threshold, our method reported 57,449 unique peptide sequences. To be noted, we kept the all the *de novo* sequences here, because in many cases multiple peptide sequences containing swapped consecutive amino acids are reported, possibly due to missing fragment peaks to distinguish them in the MS/MS spectra. For those cases, the constrained *de novo* peptide sequencing algorithm will report very similar peptides with nearly identical global matching scores.

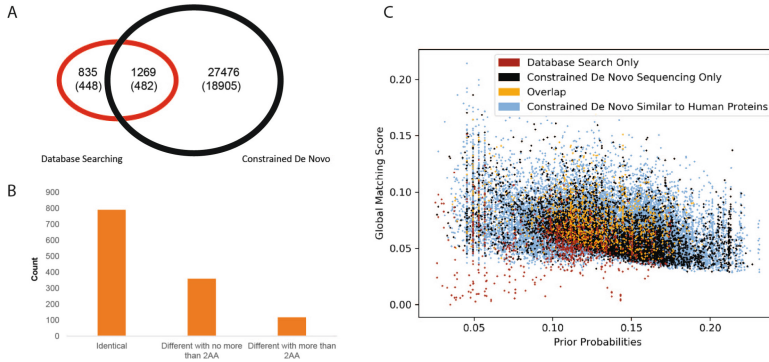


Fig. 5. (A) The comparison of PSMs and identified unique peptides (in parentheses) reported by database searching and constrained *de novo* sequencing. (B) Number of amino acids difference in overlapped IDs from database search and constrained *de novo*. (C) The prior probability and matching scores of the PSMs reported by the constrained *de novo* sequencing and database search approach. The PSMs are depicted in different colors: *orange* for those detected by both approaches, *red* for those detected by database searching only, and *black* for those detected by *de novo* sequencing only while *blue* for those reported by *de novo* sequencing and also have at least 50% sequence similarity to human proteins (Color figure online)

Comparison with Database Searching Results. MSGF+ is employed to identify peptides by searching against the human proteome database. The computation takes 1,102 min on a Linux computer (Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60 GHz). It reported 4,415 PSMs given 5% false discovery rate¹. Among these PSMs, 2,104 are identified as peptides of lengths between 9 to 12 residues (corresponding to 764 unique peptide sequences), which are putative HLA-C*0501 bounded neoepitope peptides. We compared the peptides identified by our constrained *de novo* sequencing algorithm with those identified by the database searching method in a Venn diagram shown in Fig. 5A. A total of 1,269 spectra are identified by both the database searching and the *de novo* sequencing method, among which 791 spectra were identified as identical peptides² by both methods: for 360 spectra, the peptides identified by the *de novo* sequencing method differ only in no more than two amino acid residues from the peptides identified by the database searching (where most of cases are two consecutive residues swaps); and for the remaining 118 spectra, the two identified peptides by these two methods differ in more than two residues, but share over 50% sequence similarity.

¹ We used a FDR threshold of 5% to be consistent with the original article [18]. When a more common FDR threshold 0.01 is used, much fewer (1,280) MS/MS spectra were identified, among which only 97 were identified as peptides with lengths between 9 and 12.

² Note that, here only the top-ranked peptides reported by the *de novo* sequencing algorithm were considered, and ILE and LEU are considered as identical amino acids in this comparison.

The PSMs reported by both the database searching and the *de novo* sequencing algorithm, and those reported by only one of these methods were investigated in the context of their prior probabilities and matching scores (Fig. 5C). The PSMs reported by both methods receive generally higher matching scores and comparable prior probabilities. 825 out of 835 PSMs reported only by the database searching method received a global matching score below the threshold 0.0058 used for selecting *de novo* sequencing results. The remaining ten PSMs received prior probabilities less than 0.1 (on average, prior probability is 0.05), indicating they are less likely neopeptide peptides. On the other hand, among the top-ranked 27,476 PSMs reported only by the *de novo* sequencing algorithm, 23,857 have the prior probabilities above 0.1. We further analyzed the 18,905 unique peptides from these 27,476 top-ranked PSMs. When searching against the human protein database containing 21,006 sequences from Uniprot [3] using Rapsearch2 [41], 14,658 (77.53%) peptides have 50% or higher sequence similarity with some peptides from human proteins, while 7,737 (40.93%) peptides differ at most two amino acids (i.e., a swap of two consecutive residues), including 1,910 (10.10%) identical peptides. Notably, although these identified peptides are more likely the true neopeptide peptides, some of the rest peptides may also be neopeptide peptides, e.g., those generated by novel gene splicing and fusion events, or PCPS [24].

Comparison with Current *de novo* Sequencing Methods. We attempted to compare our method with the most recently developed *de novo* sequencing method uniNovo [16] on the HLA-C peptidomic dataset. The parameters of uniNovo are chosen in consistence with the experimental settings: (1) the ion tolerance: 0.3 Da; (2) precursor ion tolerance: 100 ppm (3) fragmentation method: HCD; (4) no enzyme specificity is selected; (5) five peptide sequences per spectrum are reported; (6) minimum length of peptides: 9; and (7) minimum accuracy: 0.8. A total of 1,863 spectra are identified by uniNovo under these parameters. Most of the sequencing results are non-conclusive: only 3–6 (on average 3.1) amino acid residues were reported in these peptides, and the gaps between the residues were reported as mass intervals (e.g., a typical output of uniNovo is [406.2043]D[204.10266]QI). Because of the non-conclusive peptide sequences in uniNovo report, we did not further compare it with the results from our constrained *de novo* sequencing algorithms. We also compared our method with another up-to-date and user-friendly *de novo* sequencing software, Novor [26]. We used the default parameters of the software for comparison. In total, Novor reported 337,717 peptide-spectrum matches (PSMs), with only one top peptide for each spectrum. We note that, as Novor inherently considers only trypsin-digested peptides in the *de novo* sequencing algorithm, and most neo-epitope peptides do not have K/R at their C-termini, we limited our comparison on those top-scored tryptic-like peptides (with K/R at their C-termini) reported by our constrained *de novo* sequencing algorithm under 1% FDR. Only 2,259 spectra were identified as tryptic-like peptides by our method, the peptide sequences reported by both methods on these spectra are, however, quite different, with an average hamming distance of 5.9. When compared to the MS-GF+

results, the peptides reported by Novor have average 4.54 hamming distance, while our *de novo* results have average 3.84 hamming distance. This comparison suggests that the prior information (i.e., the PSSM) employed in the constrained sequencing algorithm helps to identify the peptide sequences that are more likely neopeptide than a generic *de novo* sequencing algorithm without using this prior information.

4 Discussion

The constrained *de novo* sequencing method was designed specifically for characterizing neopeptide sequences from their MS/MS spectra acquired in immunopeptidomic experiments. The algorithm does not rely on a database of potential neopeptide peptides, and thus can identify peptides that are not contiguous subsequences of proteins in a database, including those resulting from novel insertion, deletion, splicing or gene fusion events, or those containing mutations (e.g., in tumor cells) or those generated by *proteasome-catalyzed peptide splicing* (PCPS) [24]. The dynamic programming algorithm adopted here allows for efficient searching in the entire space of peptide sequences within a range of desirable lengths (e.g., 9–12 residues). The results showed that, when peptides can be obtained by both methods, the peptide sequence reported by the *de novo* sequencing method often match with that from database searching, with at most one swap between two consecutive amino acid residues. Notably, unlike existing *de novo* sequencing algorithms (e.g., uniNovo) often reporting many putative sequence tags each with relatively low sequence coverage of target peptide, the constrained *de novo* sequencing method report one or a few complete peptide sequence with desirable length. As a result, it is straightforward to search for the occurrence of peptide sequences in a protein database, even for those generated by PCPS (e.g., concatenated from two subpeptides in different proteins).

The results on the testing dataset showed that many MS/MS spectra that were not identified by the database searching approach were identified as putative neopeptide peptides by the constrained *de novo* sequencing algorithm. This is probably due to the fact that the constrained *de novo* sequencing method benefits from the incorporation of PSSMs as prior probabilities, which prefers the peptides with high immunogenicities (i.e., likely to be presented by MHC-I). This is consistent with the typical experimental setting in immunopeptidomics, where peptides bound to a target MHC-I protein (e.g., HLA-C for the dataset used here) are enriched before the LC-MS/MS analyses. Hence, we anticipate a majority of MS/MS spectra result from the those peptides and thus can be identified using the constrained *de novo* sequencing method. On the other hand, other peptides (not bound to the target MHC-I molecule) are not of interests in immunopeptidomics, and thus it is not a concern if the *de novo* sequencing method cannot identify them.

The PSSMs adopted in this study were constructed by using known peptide sequences bound to a target MHC-I protein (HLA-C). The PSSMs for some desirable lengths are not informative as there are only very few known peptides of the

respective length (e.g., only two known sequences have 12 residues in lengths). The PSSMs for some other classes of MHC-I may be even less characterized in current literature. We expect more accurate PSSMs can be derived after more neopeptide peptides become available with the advances of immunopeptidomic analyses, which can further improve the constrained *de novo* sequencing as presented here. Moreover, it is anticipated the preferences of MHC-I can be different in different patient because of the presence of many alleles of MHC-I encoding genes in human population. Therefore, specific PSSMs may be needed to be constructed for different MHC-I alleles so that appropriate PSSMs can be selected (based on *HLA typing* from the patient's genomic sequencing data [15,38]) for neopeptide peptide analyses of an individual patient.

The method presented here can also be applied to sequencing of other types of neopeptide peptides. For example, even though the attention has been most focused on the peptides presented by MHC-I that stimulates the cytotoxic killer T-cell responses, the peptides presented by MHC-II that are important for CD4+ helper T-cell responses [21] can also be characterized using a similar approach. The MHC-II presented peptides are typically longer in length and more variable, and thus more data are required to derive useful prior PSSM models.

Acknowledgements. This work was supported by the NIH grant 1R01AI108888 and the Indiana University Precision Health Initiative (IU-PHI).

References

1. Bhattacharya, R., Sivakumar, A., Tokheim, C., Guthrie, V.B., Anagnostou, V., Velculescu, V.E., Karchin, R.: Evaluation of machine learning methods to predict peptide binding to MHC class I proteins. *bioRxiv*, p. 154757 (2017)
2. Blum, J.S., Wearsch, P.A., Cresswell, P.: Pathways of antigen processing. *Annu. Rev. Immunol.* **31**, 443–473 (2013)
3. Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bairoch, A.: UniProtKB/Swiss-Prot: the manually annotated section of the UniProt knowledgebase. *Plant Bioinf.: Methods Protoc.* **406**, 89–112 (2007)
4. Bouvier, M., Wiley, D.C.: Importance of peptide amino and carboxyl termini to the stability of MHC class I molecules. *Science* **265**(5170), 398–402 (1994)
5. Caron, E., Kowalewski, D.J., Koh, C.C., Sturm, T., Schuster, H., Aebersold, R.: Analysis of major histocompatibility complex (MHC) immunopeptidomes using mass spectrometry. *Mol. Cell. Proteomics* **14**(12), 3105–3117 (2015)
6. Chalmers, Z.R., Connelly, C.F., Fabrizio, D., Gay, L., Ali, S.M., Ennis, R., Schrock, A., Campbell, B., Shlien, A., Chmielecki, J., et al.: Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Med.* **9**(1), 34 (2017)
7. Comber, J.D., Philip, R.: MHC class I antigen presentation and implications for developing a new generation of therapeutic vaccines. *Ther. Adv. Vaccines* **2**(3), 77–89 (2014)
8. Cottrell, J.S., London, U.: Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **20**(18), 3551–3567 (1999)

9. Dustin, M.L.: Cancer immunotherapy: killers on sterols. *Nature* **531**(7596), 583–584 (2016)
10. Editorial, N.B.: The problem with neoantigen prediction. *Nat. Biotech.* **35**(2), 97 (2017)
11. 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)
12. Eng, J.K., McCormack, A.L., Yates, J.R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* **5**(11), 976–989 (1994)
13. Flower, D.R.: Towards in silico prediction of immunogenic epitopes. *TRENDS Immunol.* **24**(12), 667–674 (2003)
14. Frank, A., Pevzner, P.: PepNovo: de novo peptide sequencing via probabilistic network modeling. *Anal. Chem.* **77**(4), 964–973 (2005)
15. Gabriel, C., Fürst, D., Faé, I., Wenda, S., Zollikofer, C., Mytilineos, J., Fischer, G.: HLA typing by next-generation sequencing-getting closer to reality. *HLA* **83**(2), 65–75 (2014)
16. Jeong, K., Kim, S., Pevzner, P.A.: UniNovo: a universal tool for de novo peptide sequencing. *Bioinformatics* **29**(16), 1953–1962 (2013)
17. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292**(2), 195–202 (1999)
18. Kaur, G., Gras, S., Mobbs, J.I., Vivian, J.P., Cortes, A., Barber, T., Kuttikkatte, S.B., Jensen, L.T., Attfield, K.E., Dendrou, C.A., et al.: Structural and regulatory diversity shape HLA-C protein expression levels. *Nat. Commun.* **8** (2017)
19. Kim, S., Pevzner, P.A.: MS-GF+ makes progress towards a universal database search tool for proteomics. *Nat. Commun.* **5** (2014)
20. Kvistborg, P., Clynes, R., Song, W., Yuan, J.: Immune monitoring technology primer: whole exome sequencing for neoantigen discovery and precision oncology. *J. Immunother. Cancer* **4**(1), 22 (2016)
21. Laidlaw, B.J., Craft, J.E., Kaech, S.M.: The multifaceted role of CD4+ T cells in CD8+ T cell memory. *Nat. Rev. Immunol.* **16**(2), 102–111 (2016)
22. Le Gallo, M., Rudd, M.L., Urick, M.E., Hansen, N.F., Zhang, S., Lozy, F., Sgroi, D.C., Vidal Bel, A., Matias-Guiu, X., Broaddus, R.R., et al.: Somatic mutation profiles of clear cell endometrial tumors revealed by whole exome and targeted gene sequencing. *Cancer* **123**, 3261–3268 (2017)
23. Li, Y.F., Arnold, R.J., Radivojac, P., Tang, H.: Protein identification problem from a Bayesian point of view. *Stat. Interface* **5**(1), 21 (2012)
24. Liepe, J., Marino, F., Sidney, J., Jeko, A., Bunting, D.E., Sette, A., Kloetzel, P.M., Stumpf, M.P., Heck, A.J., Mishto, M.: A large fraction of HLA class I ligands are proteasome-generated spliced peptides. *Science* **354**(6310), 354–358 (2016)
25. Linnemann, C., Van Buuren, M.M., Bies, L., Verdegaal, E.M., Schotte, R., Calis, J.J., Behjati, S., Velds, A., Hilkmann, H., El Atmioui, D., et al.: High-throughput epitope discovery reveals frequent recognition of neo-antigens by CD4+ T cells in human melanoma. *Nat. Med.* **21**(1), 81–85 (2015)
26. Ma, B.: Novor: real-time peptide de novo sequencing software. *J. Am. Soc. Mass Spectrom.* **26**(11), 1885–1894 (2015)
27. 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**(20), 2337–2342 (2003)

28. Neefjes, J., Jongmsma, M.L., Paul, P., Bakke, O.: Towards a systems understanding of MHC class I and MHC class II antigen presentation. *Nat. Rev. Immunol.* **11**(12), 823–836 (2011)
29. Nik-Zainal, S., Davies, H., Staaf, J., Ramakrishna, M., Glodzik, D., Zou, X., Martincorena, I., Alexandrov, L.B., Martin, S., Wedge, D.C., et al.: Landscape of somatic mutations in 560 breast cancer whole-genome sequences. *Nature* **534**(7605), 47–54 (2016)
30. Schumacher, T.N., Schreiber, R.D.: Neoantigens in cancer immunotherapy. *Science* **348**(6230), 69–74 (2015)
31. Tanner, S., Shu, H., Frank, A., Wang, L.-C., Zandi, E., Mumby, M., Pevzner, P.A., Bafna, V.: InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.* **77**(14), 4626–4639 (2005)
32. Tran, N.H., Levine, Z., Xin, L., Shan, B., Li, M.: Protein identification with deep learning: from abc to xyz. *arXiv preprint* (2017). [arXiv:1710.02765](https://arxiv.org/abs/1710.02765)
33. Tran, N.H., Zhang, X., Xin, L., Shan, B., Li, M.: De novo peptide sequencing by deep learning. *Proc. Natl. Acad. Sci.* **114**(31), 8247–8252 (2017)
34. Vacic, V., Iakoucheva, L.M., Radivojac, P.: Two sample logo: a graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* **22**(12), 1536–1537 (2006)
35. Vita, R., Overton, J.A., Greenbaum, J.A., Ponomarenko, J., Clark, J.D., Cantrell, J.R., Wheeler, D.K., Gabbard, J.L., Hix, D., Sette, A., et al.: The immune epitope database (IEDB) 3.0. *Nucleic Acids Res.* **43**(D1), D405–D412 (2014)
36. Vizcaíno, J.A., Deutsch, E.W., Wang, R., Csordas, A., Reisinger, F., Rios, D., Dianes, J.A., Sun, Z., Farrah, T., Bandeira, N., et al.: Proteomexchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* **32**(3), 223–226 (2014)
37. Wan, Y., Yang, A., Chen, T.: PepHMM: a hidden markov model based scoring function for mass spectrometry database search. *Anal. Chem.* **78**(2), 432–437 (2006)
38. Xie, C., Yeo, Z.X., Wong, M., Piper, J., Long, T., Kirkness, E.F., Biggs, W.H., Bloom, K., Spellman, S., Vierra-Green, C., et al.: Fast and accurate HLA typing from short-read next-generation sequence data with xHLA. *Proc. Natl. Acad. Sci.* 201707945 (2017)
39. Yarchoan, M., Johnson III, B.A., Lutz, E.R., Laheru, D.A., Jaffee, E.M.: Targeting neoantigens to augment antitumour immunity. *Nat. Rev. Cancer* **17**(4), 209–222 (2017)
40. Zhang, L., Udaka, K., Mamitsuka, H., Zhu, S.: Toward more accurate pan-specific MHC-peptide binding prediction: a review of current methods and tools. *Briefings Bioinf.* **13**(3), 350–364 (2011)
41. Zhao, Y., Tang, H., Ye, Y.: RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics* **28**(1), 125–126 (2011)



Reverse de Bruijn: Utilizing Reverse Peptide Synthesis to Cover All Amino Acid k -mers

Yaron Orenstein^(✉) 

Department of Electrical and Computer Engineering,
Ben-Gurion University of the Negev, Beer-Sheva, Israel
yaronore@bgu.ac.il

Abstract. Peptide arrays measure the binding intensity of a specific protein to thousands of amino acid peptides. By using peptides that cover all k -mers, a comprehensive picture of the binding spectrum is obtained. Researchers would like to measure binding to the longest k -mer possible, but are constrained by the number of peptides that can fit into a single microarray. A key challenge is designing a minimum number of peptides that cover all k -mers. Here, we suggest a novel idea to reduce the length of the sequence covering all k -mers by utilizing a unique property of the peptide synthesis process. Since the synthesis can start from both ends of the peptide template, it is enough to cover each k -mer or its reverse, and use the same template twice: in forward and reverse. Then, the computational problem is to generate a minimum length sequence that for each k -mer either contains it or its reverse. We developed an algorithm ReverseCAKE to generate such a sequence. ReverseCAKE runs in time linear in the output size and is guaranteed to produce a sequence that is longer by at most $\Theta(\sqrt{n} \log n)$ characters compared to the optimum n . The obtained saving factor by ReverseCAKE approaches the theoretical lower bound as k increases. In addition, we formulated the problem as an integer linear program and empirically observed that the solutions obtained by ReverseCAKE are near-optimal. Through this work we enable more effective design of peptide microarrays.

Keywords: de Bruijn graph · de Bruijn sequence · Peptide array
Reverse synthesis · Array design

1 Introduction

Protein-peptide interactions are a central focus of biological research. They play roles in many cellular processes. Some proteins, such as enzymes and antibodies, bind short peptides and by that affect their imminent function. Proteins bind to different peptides with variable affinities. Studying the specificity of protein-peptide binding is a fundamental goal in understanding cellular processes.

Technologies measure the binding intensity of a protein to many peptides (e.g., peptide microarrays [1–3]). These technologies synthesize a large set of amino acid peptides, and measure the binding intensity of a specific protein to each of these peptides. Some technologies use random peptide sequences [2, 3]. Others use sequences that cover all possible amino acid k -mers [1]. One way to cover all k -mers is to use *de Bruijn sequences*, which are known to be the most compact sequences to cover all k -mers [4, 5]. The length of a de Bruijn sequence of order k over alphabet $|\Sigma|$ is $|\Sigma|^k$, where the amino acid alphabet is of size $|\Sigma| = 20$. Due to the exponential dependency on k and small space on the experimental device, these technologies are limited to a small value of k (e.g. $k = 2$ [1]). Despite the universal and high-throughput nature of these technologies, the data produced are still limited. For many proteins the binding depends on more than two amino acid positions. Covering all k -mers for a greater value of k will lead to improved understanding of peptide interactions.

Here, we utilize for the first time a unique property of amino acid peptide synthesis process to generate smaller peptide libraries. As peptide synthesis can start from both the N-terminus and C-terminus [6], one can save by using this reverse property: if the synthesis starts from both ends, whenever a k -mer is included, its reverse is included as well, and there is no need to cover it again. This brings up the following question: a sequence S is called a *reverse de Bruijn sequence* of order k (RdB sequence for short) if for each k -mer either the k -mer or its reverse are included in S . Can we construct an optimal (minimum length) RdB sequence? Theoretically, if for each k -mer T the sequence S includes either T or its reverse but not both, one could save a factor of nearly 2 compared to the length of a de Bruijn sequence.

Several solutions have been suggested to generate sequence libraries that cover all possible k -mers in the most compact space possible. A de Bruijn sequence is the shortest sequence in which each k -mer appears exactly once. Its length is given by $|\Sigma|^k + k - 1$. De Bruijn sequences were used in protein binding microarrays for $k = 10$ [7]. A reduction of DNA libraries by half was achieved by utilizing the reverse-complementarity property of double-stranded DNA [8–10]. Other methods produce compact unstructured RNA libraries to measure protein-RNA binding [11, 12]. But, none of those studies considered the property of reverse peptide synthesis, i.e. the need to cover each k -mer by itself or its reverse.

In this study we address the problem of constructing a compact RdB sequence. We take the view point of a sequence as a path in a de Bruijn graph, where an RdB sequence and its reverse are two reverse paths. We first give a lower bound for the length of an RdB sequence. Then, we give a sufficient and necessary condition for a de Bruijn graph to represent two reverse RdB sequences. As a consequence, we prove that a minimum length RdB cannot achieve the lower bound due to palindromes. We present a linear time near-optimal algorithm, ReverseCAKE (Reverse Covering All K -mErs), to make a de Bruijn graph obtain these properties. Once a de Bruijn graph obtains these properties, a modified Euler tour algorithm can run on it to produce the sequence. Moreover, we

formulate the problem as an integer linear program (ILP). We implemented the algorithm and the ILP formulation and we demonstrate the savings they achieve. The results enable saving a factor of almost two compared to using a regular de Bruijn sequence. The code and software are freely available from <https://github.com/yaronore/reverse-de-bruijn>.

2 Preliminaries

A *directed graph* (digraph or simply a graph) $G = (V, E)$ is a set of vertices $V = \{v_1, v_2, \dots, v_n\}$ and a set of edges $E = \{e_1, e_2, \dots, e_m\}$. Each edge is an ordered pair of vertices (v_i, v_j) , and we say the edge is directed from v_i to v_j . The *indegree* of vertex v is the number of edges entering v . Similarly, the *outdegree* is the number of edges outgoing from v . A vertex is *balanced* if its indegree equals its outdegree. A *path* in a digraph is a sequence of vertices, v_{i_1}, \dots, v_{i_k} , such that for each $1 \leq j < k$ there is an edge $(v_{i_j}, v_{i_{j+1}})$. A *cycle* is a path where $i_1 = i_k$. A digraph is *strongly connected* if for every pair of vertices u, v there exists a path from u to v and a path from v to u .

An *Eulerian tour* through a digraph G is a cycle that traverses all edges in G , such that each edge is traversed exactly once. If a digraph contains an Eulerian tour, we call it *Eulerian*. A digraph is Eulerian if and only if it is strongly connected and all vertices are balanced [13].

A *de Bruijn sequence* of order k over alphabet Σ is a minimum length sequence that covers each k -mer over Σ exactly once. For convenience, we define the *length* of the sequence as the number of k -mers in it. Hence, a sequence of length t contains $t + k - 1$ characters, or t characters if it is cyclic. A de Bruijn sequence has length $|\Sigma|^k$, which is the minimum possible for covering all k -mers.

A *de Bruijn graph* of order k is a digraph in which for every possible k -mer x_1, \dots, x_k there is a vertex denoted by $[x_1, \dots, x_k]$. An edge may exist from u to v if $u = [x_1, \dots, x_k]$ and $v = [x_2, \dots, x_{k+1}]$. Each edge represents a unique $(k + 1)$ -mer. For example, the edge (u, v) above represents (x_1, \dots, x_{k+1}) . To distinguish vertices from edges, we will use square brackets for vertices. Hence, (x_1, \dots, x_{k+1}) is the edge between $[x_1, \dots, x_k]$ and $[x_2, \dots, x_{k+1}]$. In a *complete* de Bruijn graph all possible edges exist, each exactly once. Consequently, for each vertex v the indegree and outdegree are $|\Sigma|$, and the graph is strongly connected. Thus, a complete de Bruijn graph is Eulerian. Any Eulerian tour represents a de Bruijn sequence of order $k + 1$.

The *reverse* of sequence (x_1, \dots, x_k) , denoted $R(x_1, \dots, x_k)$, is defined as the sequence obtained by reversing the original sequence, i.e. $R(x_1, \dots, x_k) = (x_k, \dots, x_1)$. For example, $R(CGAA) = AAGC$. A sequence s is called a *palindromic* sequence, or in short a *palindrome*, if $s = R(s)$. For example, $ACCA$ is a palindrome. An *homomorphic k -mer* is composed of a single letter, e.g. $AA \dots A$.

We define a *reverse de Bruijn sequence* of order k over alphabet Σ (RdB sequence for short) as a sequence such that for each k -mer s , at least one of s and $R(s)$ are in the sequence. Note that unlike a regular de Bruijn sequence, the definition of an RdB sequence does not require minimality. An RdB sequence is *optimal* if it is of minimum length. An RdB sequence is cyclic, and can be easily turned to a linear sequence by appending the first $k - 1$ characters.

Given a directed path F in a de Bruijn graph, its *reverse path* is defined as the path R in which each edge (u, v) in F is replaced by the edge $(R(v), R(u))$. For example, for the path $ACG \rightarrow CGG \rightarrow GGT$, its reverse is $TGG \rightarrow GGC \rightarrow GCA$ (see Fig. 1). We will refer to F and R as forward and reverse paths, respectively.

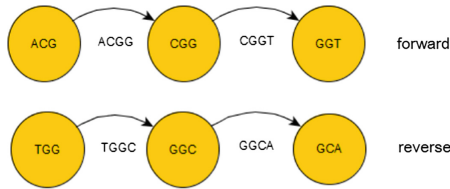


Fig. 1. An illustration of forward and reverse paths (top and bottom, respectively). The forward path traverses the edges in their direction. The corresponding reverse path traverses the reverse edges in reverse direction.

3 Results

3.1 A Lower Bound for the Length of an RdB Sequence

We derive a lower bound for the length of an RdB sequence from k -mer counts.

Proposition 1. Denote $n(k)$ the length of an optimal RdB sequence of order k .

$$n(k) \geq \frac{1}{2} \cdot (|\Sigma|^k + |\Sigma|^{\lfloor (k+1)/2 \rfloor}) \quad (1)$$

Proof. We derive the lower bound by counting palindromic and non-palindromic edges. It depends on the number of k -mers that are palindromes, since each palindrome must be represented by itself, while each non-palindromic k -mer can be represented by either itself or its reverse. For even k the first $\frac{k}{2}$ characters define the last $\frac{k}{2}$ characters of a palindrome. For odd k , the first $\frac{k-1}{2}$ characters define the last $\frac{k-1}{2}$, and the middle character can be any letter. Hence, there are exactly $|\Sigma|^{\lfloor (k+1)/2 \rfloor}$ different palindromes. In total, counting all palindromes and half of all non-palindromes gives $n(k) \geq \frac{1}{2} \cdot (|\Sigma|^k - |\Sigma|^{\lfloor (k+1)/2 \rfloor}) + |\Sigma|^{\lfloor (k+1)/2 \rfloor}$.

3.2 A de Bruijn Graph Representing Two Reverse RdB Sequences

We give a sufficient and necessary condition for a de Bruijn graph to represent two reverse RdB sequences. This will be useful to prove that no RdB can achieve the lower bound. It will also be relevant for a de Bruijn graph edge augmentation we show below, as it will make a complete de Bruijn graph obtain these properties. We take the viewpoint of a sequence represented as a path in a de Bruijn graph. We first prove the following lemma:

Lemma 1. *For every incoming non-homomorphic edge e into a palindromic vertex, there is a unique outgoing edge e' such that $e' = R(e)$.*

Proof. Denote the vertex label as $v = [x_1, \dots, x_k]$, which is equal to $[x_k, \dots, x_1]$ as it is a palindromic vertex. Denote an incoming edge by $e = (y, x_1, \dots, x_k)$. Its reverse is $R(e) = (x_k, \dots, x_1, y)$, which is an outgoing edge from v . $e = R(e)$ if and only if e is homomorphic.

Theorem 1. *The set of edges of de Bruijn graph G represents two reverse RdB sequences \iff de Bruijn graph G has the following properties:*

1. *All vertices in G are balanced.*
2. *G is strongly connected.*
3. *There is a perfect matching of edges and their reverse in G .*
4. *All palindromic vertices in G have an even in and outdegree (disregarding homomorphic edges).*

Proof. \rightarrow Each k -mer in a sequence is an edge in the graph. As an RdB sequence and its reverse are cyclic each vertex is entered and exited the same number of times, and it follows that the vertices are balanced. As an RdB sequence and its reverse cover all k -mers, all possible edges exist and it follows that the graph is strongly connected. The edges of the paths representing the RdB sequence and its reverse are in perfect matching of reverse edges with each other by definition of forward and reverse paths. Last, by Lemma 1 each palindromic vertex is entered and exited by both paths at the same time. Thus, it must be entered and exited an even number of times (disregarding homomorphic edges whose traversal does not change the paths' location).

\leftarrow Given that de Bruijn graph G has the four properties, it has two reverse paths that cover all of its edges. The paths enter and exit each vertex the same number of times as the vertices are balanced. The paths cover together all edges as the graph is strongly connected. The perfect matching is necessary to have two reverse paths in the graph. Last, by Lemma 1 when the paths enter the same vertex (palindromic vertices) they never reach a dead-end as there is an even number of outgoing and incoming edges (disregarding homomorphic edges whose traversal does not change the paths' location).

As a consequence, no RdB sequence can achieve the lower bound:

Corollary 1. *There is no RdB sequence that achieves the lower bound.*

Proof. Assume, contrary to the claim, that there is a reverse Bruijn sequence that achieves the lower bound. Thus, the sequence and its reverse path are two edge-disjoint paths in an augmented de Bruijn graph, where each original palindromic edge is doubled and all other edges appear only once. The augmented graph has vertices with unequal indegree and outdegree due to the augmentation of palindromic edges, contradicting Theorem 1.

Given a graph with the listed properties we can apply an algorithm to find two reverse paths that cover all edges. The algorithm is based on a modification of the Euler tour algorithm [13] run on an augmented de Bruijn graph. The algorithm for generating the sequence will work on an augmented de Bruijn graph of order $k - 1$. We previously presented it [10] and repeat it here for sake of clarity.

Algorithm 1. Find forward and reverse paths that cover all edges in an augmented de Bruijn graph $G = (V, E)$ of order $k - 1$.

1. Initially all edges are unmarked, $\mathcal{F} = \mathcal{R} = \emptyset$, and $A = \{u\}$, an arbitrary vertex.
 2. While $A \neq \emptyset$ do
 3. $F = R = \emptyset$. Pick any starting vertex $v = [x_1, \dots, x_{k-1}]$ from A .
 4. While there exists an unmarked edge $e = (x_1, \dots, x_k)$ outgoing from v do
 5. Append e to F . Prepend $R(e)$ to R .
 6. Mark e and $R(e)$.
 7. Set $v = [x_2, \dots, x_k]$; $A = A \cup \{v\}$.
 8. Remove v from A .
 9. If $F \neq \emptyset$, add F to \mathcal{F} ; add R to \mathcal{R} ;
 10. Merge the cycles in \mathcal{F} to obtain a single forward path. Do the same for \mathcal{R} .
-

Theorem 2. *Algorithm 1 returns in $O(|V|)$ time forward and reverse paths that cover together all edges of the augmented graph and represent two RdB sequences.*

The algorithm and its proof are similar to that of a modified Euler tour algorithm we previously presented [10] (and will not be repeated here). We first show that if the forward path F reaches a dead-end, then so does the reverse path R , and in that case a cycle is closed (note that each pair F, R constructed in steps 4–7 are reverse paths by the way they are constructed). Then, we show that the cycles in \mathcal{F} can be merged into one cycle. Third, we deduce that a strongly connected component is covered by \mathcal{F} and \mathcal{R} . Last, we conclude that \mathcal{F} and \mathcal{R} cover all edges, since there is only one strongly connected component in any de Bruijn graph. The only difference between the proofs is the case where both paths enter the same vertex simultaneously and reach a dead-end.

Lemma 2. *If the forward traversal reaches a dead-end at a palindromic vertex, then so does the reverse at the same vertex. Both paths close a cycle in this case.*

Proof. Recall that when F reaches a palindromic vertex R must reach it a well, and this is the only case where both paths reach a vertex together. By Lemma 1 a palindromic vertex has even in and outdegrees (excluding homomorphic edges). Denote by (x_1, \dots, x_k) an incoming edge used by F . Then, the reverse outgoing edge, which is traversed by R , is (x_k, \dots, x_1) . As it is a palindromic vertex, or equivalently from the fact that both reach the vertex simultaneously, we get that $[x_2, \dots, x_k] = [x_k, \dots, x_2]$. It follows that in all traversals of this vertex F and R reach the vertex simultaneously. Hence, when F reaches a dead-end, all incoming and outgoing edges were already traversed, and they are all of the form (a, x_2, \dots, x_n) and (x_n, \dots, x_2, a) , $\forall a \in \Sigma$. For each traversal of such pair of incoming edges, a pair of outgoing edges is traversed. Thus, if the traversal ends at the vertex, it must be that the traversal started from that vertex (otherwise, there would have been unmarked outgoing edges to traverse). In other words, both paths close a cycle in this case.

3.3 Constructing a Near-Optimal RdB Sequence in Linear Time

In this approach, for each palindromic edge, we add to a complete de Bruijn graph all possible cyclic shifts of it. More formally, for even k let $k = 2l$. For the palindrome $e = (x_1, \dots, x_l, x_l, \dots, x_1)$ we add k edges corresponding to all possible cyclic shifts of e . Similarly, for odd k let $k = 2l + 1$ we add all cyclic shifts of $(x_1, \dots, x_l, x_{l+1}, x_l, \dots, x_1)$. Obviously, since these edges form a cycle, all vertices remain balanced. The added edges match in reverse pairs. For each edge that represents the cyclic shift starting at position i , for $1 < i < \lfloor (k+1)/2 \rfloor$, the matching edge starts at $k+2-i$. Hence, a perfect matching exists after adding the new cycles. For even k , unless the k -mer is homomorphic, this cycle contains two edges that are palindromes, $(x_1, \dots, x_l, x_l, \dots, x_1)$ and $(x_l, \dots, x_1, x_1, \dots, x_l)$, so only one cycle is added for both, and the cycle doubles both palindromic edges. In total, during the edge augmentation process, for each palindromic k -mer we add at most k edges. For example, for the palindromes $AGGA$ and $GAAG$ we add $AGGA$, $GGAA$, $GAAG$ and $AAGG$ (see Fig. 2). The added edges $GGAA$ and $AAGG$ match each other as a reverse edge pair. The added palindromes match the original edges in the graph. The resulting augmented graph contains at most $|\Sigma|^k + k \cdot |\Sigma|^{\lfloor (k+1)/2 \rfloor}$ edges, where the first term is the number of edges in the original de Bruijn graph and the second is k edges for each palindrome.

In some cases, the number of added edges can be reduced. If the palindrome (x_1, \dots, x_k) is periodic, then the number of cyclic shifts needed to return to the original k -mer is the length of the period. For example, the period of $ACCAACCA$ is 4. Only four edges suffice in this case, the edges $ACCAACCA$, $CCAACCAA$, $CAACCAAC$ and $AACCAACC$. So, each periodic palindrome requires an addition of the number of edges equal to the length of its period. Hence, a smaller augmented graph and a shorter RdB sequence can be obtained by considering the different possible periods.

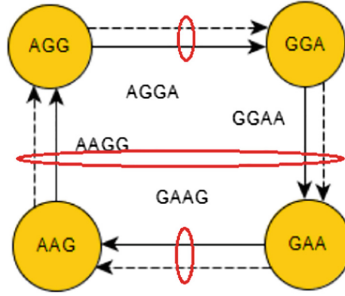


Fig. 2. A cycle and edge matching. For the pair of palindromes *AGGA* and *GAAG*, all cyclic shifts of these palindromes are added once (dashed edges). In the matching, palindromic edges in the original cycle are paired with their added copies (encircled by small red ovals). Other non-palindromic added edges are paired (encircled by a large red oval). (Color figure online)

At the end of the above augmentation, an additional augmentation is required to palindromic vertices with odd degrees. All palindromic vertices must have an even in and outdegrees (disregarding homomorphic edges) by Lemma 1. To make sure all palindromic vertices have even degrees following the above augmentation, all palindromic vertices with odd degrees are matched in pairs. Then, $k - 1$ edges connecting them in a cycle are added. This augmentation preserves degree balance, graph connectivity, perfect matching of reverse edges and does not affect degree of other palindromic vertices. Since there are at most $|\Sigma|^{\lfloor k/2 \rfloor}$ palindromic vertices, in this process at most $(k - 1)|\Sigma|^{\lfloor k/2 \rfloor}$ edges are added.

Algorithm 1 produces two sequences, forward and reverse, each of which is an RdB sequence (Fig 3), in time linear in the size of the graph [10]. The length of each of the produced sequences is the number of edges divided by two. For each palindromic edge at most k edges were added. For each palindromic vertex 0 or $k - 1$ edges were added. So, the total length of the sequence is bounded by $(|\Sigma|^k + k|\Sigma|^{\lfloor (k+1)/2 \rfloor} + (k - 1)|\Sigma|^{\lfloor k/2 \rfloor})/2$. This is an addition of $\Theta(\sqrt{L} \log(L))$ characters, where L denotes the lower bound in Proposition 1 for an RdB sequence of order k . We call the augmentation process followed by Algorithm 1 ReverseCAKE.

Figure 4A (and Table 1 in the Appendix) show the results of ReverseCAKE for different values of k . As we can see, the sequence obtained by ReverseCAKE is of length nearly half that of the original de Bruijn sequence. For example, for $k = 4$ and amino acid alphabet, it is within 1 percent of $20^4/2$ and within 220 characters from the lower bound.

3.4 Integer Linear Programming Formulation

We present an ILP formulation to calculate the minimum length RdB sequence. There are $|\Sigma|^k$ integer variables X_i . Each X_i corresponds to the number of times the k -mer occurs in the sequence.

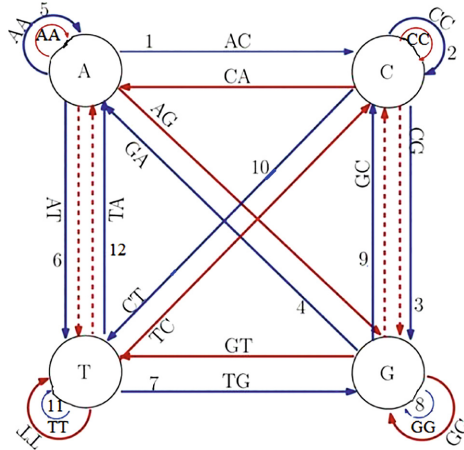


Fig. 3. An augmented de Bruijn graph of order 1 and an example of forward and reverse paths in it. Palindromic edges AA , CC , GG and TT were added first as cyclic shifts of all palindromes. Then, dashed edges AT , TA , CG and GC were added to turn odd degree palindromic vertices to even degree. The blue and brown paths represent the forward and reverse paths, respectively. Numbers on edges are the order of the edges in the forward path. The sequences are $ACCGAATGGCTT$ and $TTCGGTAAGCCA$ for forward and reverse paths, respectively. (Color figure online)

As we aim for the shortest sequence, the objective function is

$$\min \sum_{i=1}^{|\Sigma|^k} X_i \tag{2}$$

The first constraint is the coverage constraint, which requires that all k -mers occur in the sequence as themselves or their reverse. Let $R(i)$ denote the reverse of k -mer i , where we use the integer representation of a k -mer as a number in radix $|\Sigma|$.

$$X_i + X_{R(i)} \geq 1 \quad 1 \leq i \leq |\Sigma|^k \tag{3}$$

The second constraint guarantees that the k -mer occurrences can form a (cyclic) sequence. We require that for each $(k - 1)$ -mer the number of k -mers with that $(k - 1)$ -mer in their suffix is equal to the number of k -mers with that $(k - 1)$ -mer in their prefix (equivalent to a flow conservation constraint). Denote $p_x(i)$ and $s_x(i)$ the x -long prefix and suffix of i , respectively.

$$\sum_{s_{k-1}(i')=i} X_{i'} = \sum_{p_{k-1}(i')=i} X_{i'} \quad 1 \leq i \leq |\Sigma|^{k-1} \tag{4}$$

We compared the memory usage and runtime of ReverseCAKE and the ILP solver. The results are summarized in Fig. 4B and C (and Table 2 in the Appendix). We used Gurobi ILP 7.5.2 solve to solve the ILP formulation [14].

Running times and memory usages were benchmarked on a single CPU of a 20-CPU Intel Xeon E5-2650 (2.3 GHz) machine with 384 GB 2133 MHz RAM. In all runs reported, the ILP solver reached an optimal solution. As expected, the ILP solver requires much more time and memory. Our linear time algorithm produces a sequence that is only negligibly longer, but in much shorter times and using less memory.

4 Summary and Discussion

We studied the problem of constructing a minimum length sequence that covers each k -mer by itself or its reverse. The problem has applications in constructing dense amino acid peptide arrays for measuring protein-peptide interactions [1–3]. Using our solution researchers will be able to improve the utilization of peptide arrays in high-throughput, universal and unbiased measurement of peptide interactions.

The problem is challenging due to palindromes, which are the reverse of themselves and must appear in any sequence. We present a linear time near-optimal algorithm ReverseCAKE to solve it. In practice, the length of the sequence produced by the algorithm nearly halves the total length of the sequence. It is very close to the optimum as empirically shown by the integer linear programming solutions. We believe that our results are of theoretical interest, and are applicable in current and future technologies that require complete coverage of amino acid k -mers under harsh space constraints.

Our study raises several open questions. Can one construct an optimal RdB sequence in polynomial time? Second, what is the number of different optimal RdB sequences? Third, can one design an optimal RdB sequence with improved coverage of gapped k -mers in cases of gapped peptide interactions? Fourth, is there a closed formula for the length of an optimal RdB sequence? Finally, in current technologies, the de Bruijn (or RdB) sequence is cut into probes of length p with overlap $k - 1$. There is no constraint that forces these probes to come from a single sequence. What is the minimum number of sequences of length p that cover all k -mers, each k -mer by itself or its reverse? Since our solution for an RdB sequence covers a few k -mers more than once (as shown by the gap between the theoretical lower bound and our solutions), a direct design of probe sequences of length p might be able to reduce the number of probes needed to cover all k -mers.

Appendix

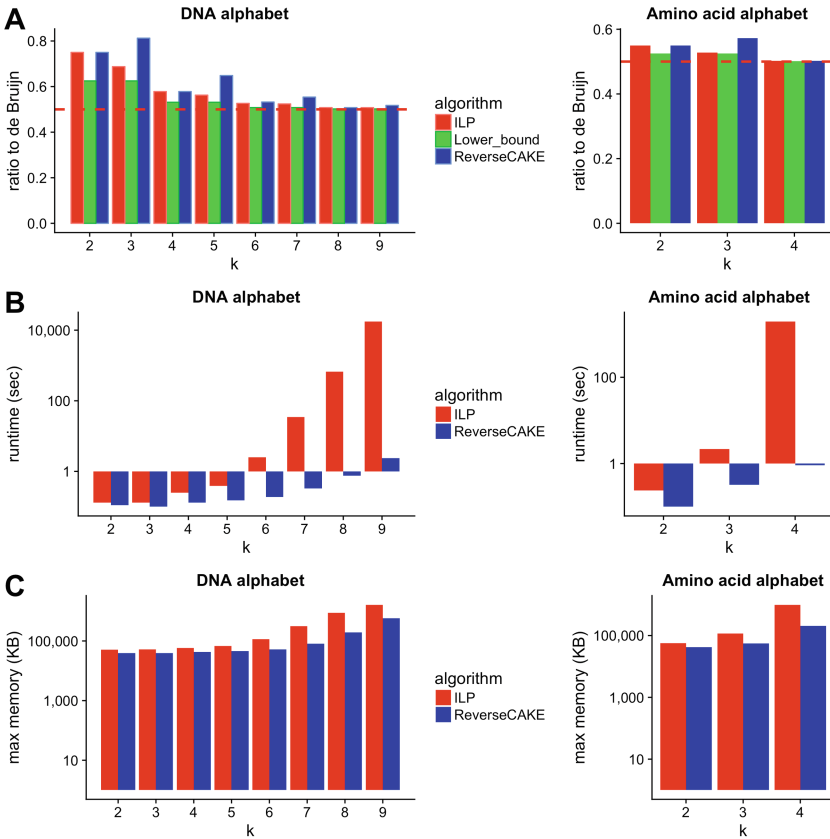


Fig. 4. Results and performance of ReverseCAKE and the ILP solver. (A) Results of sequence lengths are portrayed as ratios to an original de Bruijn sequence ($|\Sigma|^k$). The lower bound is from Proposition 1. The dashed red line is at half. (B,C) Runtimes and maximum memory usage of the algorithms, respectively. Y-axis is on a log-scale. (Color figure online)

Table 1. Lengths of reverse de Bruijn sequences produced by ReverseCAKE and an ILP solver. The columns are organized as follows: (i) the alphabet, where aa stands for amino acid; (ii) the length of a regular de Bruijn sequence that does not exploit reverse peptide synthesis; (iii) the lower bound on RdB sequence length (Proposition 1); (iv–v) the lengths of the sequence computed by ReverseCAKE (Sect. 3.3) and an ILP solver that reached an optimal solution (Sect. 3.4); (vi) the saving factor is the ratio between an optimal RdB sequence and a de Bruijn sequence.

k	Alphabet	de Bruijn	Lower bound	ReverseCAKE	ILP (optimal)	Saving factor
2	DNA	16	10	12	12	0.75
3	DNA	64	40	52	44	0.69
4	DNA	256	136	148	148	0.58
5	DNA	1,024	544	664	576	0.56
6	DNA	4,096	2,080	2,180	2,156	0.53
7	DNA	16,384	8,320	9,076	8,584	0.52
8	DNA	65,536	32,896	33,276	33,276	0.51
9	DNA	262,144	131,584	135,628	133,064	0.51
2	aa	400	210	220	220	0.55
3	aa	8,000	4,200	4,580	4,220	0.53
4	aa	160,000	80,200	80,420	80,420	0.50

Table 2. Performance evaluation of ReverseCAKE and an ILP solver. The runtime and maximum memory usage are reported in seconds (sec) and kilobytes (KB).

k	Alphabet	ReverseCAKE (sec)	ReverseCAKE (KB)	ILP (sec)	ILP (KB)
2	DNA	0.11	39,456	0.13	50,608
3	DNA	0.10	38,964	0.13	52,496
4	DNA	0.13	42,456	0.25	58,308
5	DNA	0.15	45,928	0.39	67,236
6	DNA	0.19	52,036	2.50	114,118
7	DNA	0.33	81,476	34.43	310,752
8	DNA	0.76	191,828	665.15	875,740
9	DNA	2.38	579,220	17,592.82	1,622,996
2	aa	0.10	41,804	0.24	56,800
3	aa	0.32	55,336	2.18	115,272
4	aa	0.92	203,548	1,963.64	982,124

References

1. Gurard-Levin, Z.A., Kilian, K.A., Kim, J., Bähr, K., Mrksich, M.: Peptide arrays identify isoform-selective substrates for profiling endogenous lysine deacetylase activity. *ACS Chem. Biol.* **5**(9), 863–873 (2010)

2. Buus, S., Rockberg, J., Forsström, B., Nilsson, P., Uhlen, M., Schafer-Nielsen, C.: High-resolution mapping of linear antibody epitopes using ultrahigh-density peptide microarrays. *Mol. Cell. Proteomics* **11**(12), 1790–1800 (2012)
3. Halperin, R.F., Stafford, P., Johnston, S.A.: Exploring antibody recognition of sequence space through random-sequence peptide microarrays. *Mol. Cell. Proteomics* **10**(3), M110.000786 (2011)
4. Berger, M.F., Philippakis, A.A., Qureshi, A.M., He, F.S., Estep III, P.W., Bulyk, M.L.: Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nat. Biotechnol.* **24**(11), 1429 (2006)
5. Fordyce, P.M., Gerber, D., Tran, D., Zheng, J., Li, H., DeRisi, J.L., Quake, S.R.: De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis. *Nat. Biotechnol.* **28**(9), 970–975 (2010)
6. Benoiton, N.L.: *Chemistry of Peptide Synthesis*. CRC Press (2016)
7. Philippakis, A.A., Qureshi, A.M., Berger, M.F., Bulyk, M.L.: Design of compact, universal DNA microarrays for protein binding microarray experiments. *J. Comput. Biol.* **15**(7), 655–665 (2008)
8. D’Addario, M., Kriege, N., Rahmann, S.: Designing q-Unique DNA sequences with integer linear programs and Euler tours in de Bruijn graphs. In: *OASICS-OpenAccess Series in Informatics*, vol. 26. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2012)
9. Smith, R.P., Riesenfeld, S.J., Holloway, A.K., Li, Q., Murphy, K.K., Feliciano, N.M., Orecchia, L., Oksenberg, N., Pollard, K.S., Ahituv, N.: A compact, in vivo screen of all 6-mers reveals drivers of tissue-specific expression and guides synthetic regulatory element design. *Genome Biol.* **14**(7), 1 (2013)
10. Orenstein, Y., Shamir, R.: Design of shortest double-stranded DNA sequences covering all k -mers with applications to protein-binding microarrays and synthetic enhancers. *Bioinformatics* **29**(13), i71–i79 (2013)
11. Orenstein, Y., Berger, B.: Efficient design of compact unstructured RNA libraries covering all k -mers. *J. Comput. Biol.* **23**(2), 67 (2016)
12. Ray, D., Kazan, H., Cook, K.B., Weirauch, M.T., Najafabadi, H.S., Li, X., Gueroussov, S., Albu, M., Zheng, H., Yang, A., et al.: A compendium of RNA-binding motifs for decoding gene regulation. *Nature* **499**(7457), 172 (2013)
13. West, D.B., et al.: *Introduction to Graph Theory*, vol. 2. Prentice Hall, Upper Saddle River (2001)
14. Gurobi Optimization, I.: Gurobi optimizer reference manual (2016). <http://www.gurobi.com>



Circular Networks from Distorted Metrics

Sebastien Roch^(✉) and Kun-Chieh Wang

Department of Mathematics, University of Wisconsin-Madison, Madison, USA
roch@math.wisc.edu

Abstract. Trees have long been used as a graphical representation of species relationships. However complex evolutionary events, such as genetic reassortments or hybrid speciations which occur commonly in viruses, bacteria and plants, do not fit into this elementary framework. Alternatively, various network representations have been developed. Circular networks are a natural generalization of leaf-labeled trees interpreted as split systems, that is, collections of bipartitions over leaf labels corresponding to current species. Although such networks do not explicitly model specific evolutionary events of interest, their straightforward visualization and fast reconstruction have made them a popular exploratory tool to detect network-like evolution in genetic datasets. Standard reconstruction methods for circular networks, such as Neighbor-Net, rely on an associated metric on the species set. Such a metric is first estimated from DNA sequences, which leads to a key difficulty: distantly related sequences produce statistically unreliable estimates. This is problematic for Neighbor-Net as it is based on the popular tree reconstruction method Neighbor-Joining, whose sensitivity to distance estimation errors is well established theoretically. In the tree case, more robust reconstruction methods have been developed using the notion of a distorted metric, which captures the dependence of the error in the distance through a radius of accuracy. Here we design the first circular network reconstruction method based on distorted metrics. Our method is computationally efficient. Moreover, the analysis of its radius of accuracy highlights the important role played by the maximum incompatibility, a measure of the extent to which the network differs from a tree.

Keywords: Phylogenetic networks · Circular networks
Finite metrics · Split decomposition · Distance-based reconstruction
Distorted metrics

1 Introduction

Trees have long been used to represent species relationships [1–3]. The leaves of a phylogenetic tree correspond to current species while its branchings indicate past speciation events. However, complex evolutionary events, such as genetic reassortments or hybrid speciations, do not fit into this elementary framework. Such non-tree-like events play an important role in the evolution of viruses,

bacteria and plants. This issue has led to the development of various notions of *phylogenetic networks* [4].

A natural generalization of phylogenetic trees is obtained by representing them as split networks, that is, collections of bipartitions over the species set. On a tree whose leaves are labeled by species names, each edge can be thought of as a bipartition over the species: removing the edge produces exactly two connected components. In this representation, trees are characterized by the fact that their splits have a certain compatibility property [5]. More generally, circular networks relax this compatibility property, while retaining enough structure to be useful as representations of evolutionary history [6]. Such networks are widely used in practice. Although they do not explicitly model specific evolutionary events (see, e.g., [7] for a discussion), their straightforward visualization and fast reconstruction have made them a popular exploratory tool to detect network-like evolution in genetic datasets [8]. They are also useful in cases where data is insufficient to single out a unique tree-like history, but instead supports many possible evolutionary scenarios.

Standard reconstruction methods for circular networks, such as the Neighbor-Net algorithm introduced in [9], rely on a metric on the species set. Such a metric, which quantifies how far apart species are in the Tree of Life, is estimated from genetic data. Very roughly, it counts how many mutations separate any two species. This leads to a key difficulty: under standard stochastic models of DNA evolution, distantly related sequences are known to produce statistically unreliable distance estimates [10,11]. This is problematic for Neighbor-Net, in particular, as it is based on the popular tree reconstruction method Neighbor-Joining, whose sensitivity to distance estimation errors is well established theoretically [12].

In the tree case, more robust reconstruction methods were developed using the notion of a distorted metric which captures the dependence of the error in the distance through a radius of accuracy [13,14]. A key insight to come out of this line of work, starting with the seminal results of [10,11], is that a phylogenetic tree can be reconstructed using only a subset of the pairwise distances—those less than roughly the chord depth of the tree. Here the chord depth of an edge is the shortest path between two leaves passing through that edge and the chord depth of the tree is the maximum depth among its edges. This result is remarkable because, in general, the depth can be significantly smaller than the diameter. As a consequence, a number of results have been obtained showing that, under common stochastic models of sequence evolution, a polynomial amount of data suffices to reconstruct a phylogenetic tree with bounded branch lengths. See e.g. [15–18]. This approach has also inspired practical reconstruction methods [19,20].

Here we design the first reconstruction method for circular networks based on distorted metrics. In addition to generalizing the chord depth, we show that, unlike the tree case, pairwise distances within the chord depth do not in general suffice to reconstruct these networks. We introduce the notion of maximum incompatibility, a measure of the extent to which the network differs from a tree,

to obtain a tight (up to a constant) bound on the required radius of accuracy. Before stating our main results, we provide some background on split networks.

2 Background

We start with some basic definitions. See [4] for an in-depth exposition.

Definition 1 (Split networks [6]). A split $S = (S_1, S_2)$ on a set of taxa \mathcal{X} is an unordered bipartition of \mathcal{X} into two non-empty, disjoint sets: $S_1, S_2 \in \mathcal{X}$, $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = \mathcal{X}$. We say that $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ is a **weighted split network** (or *split network for short*) on a set of \mathcal{X} if \mathcal{S} is a set of splits on \mathcal{X} and $w : \mathcal{S} \rightarrow (0, \infty)$ is a positive split weight function. We assume that any two splits $S^{(1)} = \{S_1^{(1)}, S_2^{(1)}\}$, $S^{(2)} = \{S_1^{(2)}, S_2^{(2)}\}$ in \mathcal{S} are distinct, that is, $S_1^{(1)} \neq S_1^{(2)}, S_2^{(1)}$.

For any $x, y \in \mathcal{X}$, we let $\mathcal{S}|_{x,y}$ be the collection of splits in \mathcal{S} separating x and y , that is,

$$\mathcal{S}|_{x,y} = \{S \in \mathcal{S} : \delta_S(x, y) = 1\},$$

where $\delta_S(x, y)$, known as the split metric, is the indicator of whether $S = (S_1, S_2)$ separates x and y

$$\delta_S(x, y) = \begin{cases} 0, & \text{if } x, y \in S_1 \text{ or } x, y \in S_2. \\ 1. & \text{otherwise.} \end{cases} \tag{1}$$

For a split $S \in \mathcal{S}|_{x,y}$, we write $S = \{S_x, S_y\}$ where $x \in S_x$ and $y \in S_y$. For simplicity, we assume that $\mathcal{S}|_{x,y} \neq \emptyset$ for all $x, y \in \mathcal{X}$. (Taxa not separated by a split can be identified.)

Let $T = (V, E)$ be a binary tree with leaf set \mathcal{X} and non-negative edge weight function $w : E \rightarrow [0, +\infty)$. We refer to $\mathcal{T} = (\mathcal{X}, V, E, w)$ as a phylogenetic tree. Any phylogenetic tree can be represented as a weighted split network. For each edge $e \in E$, define a split on \mathcal{X} as follows: after deleting e , the vertices of \mathcal{T} form two disjoint connected components with corresponding leaf sets S^1 and S^2 ; we let $S_e = \{S^1, S^2\}$ be the split generated by e in this way. Conversely, one may ask: given a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$, is there a phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$ such that $\mathcal{S} = \{S_e : e \in E\}$ (with $w(S_e) = w(e)$)? To answer this question, we need the concept of compatibility.

Definition 2 (Compatibility [21]). Two splits $S^{(1)} = \{S_1^{(1)}, S_2^{(1)}\}$ and $S^{(2)} = \{S_1^{(2)}, S_2^{(2)}\}$ are called **compatible**, if at least one of the following intersections is empty:

$$S_1^{(1)} \cap S_1^{(2)}, \quad S_1^{(1)} \cap S_2^{(2)}, \quad S_2^{(1)} \cap S_1^{(2)}, \quad S_2^{(1)} \cap S_2^{(2)}.$$

We write $S^{(1)} \sim S^{(2)}$ to indicate that $S^{(1)}$ and $S^{(2)}$ are compatible. Otherwise, we say that the two splits are **incompatible**. A set of splits \mathcal{S} is called **compatible** if all pairs of splits in \mathcal{S} are compatible.

In words, for any two splits, there is one side of one and one side of the other that are disjoint. The following result was first proved in [21]. Given a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$, there is a phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$ such that $\mathcal{S} = \{S_e : e \in E\}$ if and only if \mathcal{S} is compatible. For a collection of splits $S^{(1)}, \dots, S^{(\ell)}$ on \mathcal{X} , we let

$$\mathcal{C}_{\mathcal{N}}(S^{(1)}, \dots, S^{(\ell)}) = \{S \in \mathcal{S} : S \sim S^{(i)}, \forall i\}, \quad (2)$$

be the set of splits of \mathcal{N} compatible with all splits in $S^{(1)}, \dots, S^{(\ell)}$, and we let

$$\mathcal{I}_{\mathcal{N}}(S^{(1)}, \dots, S^{(\ell)}) = \{S \in \mathcal{S} : \exists i, S \not\sim S^{(i)}\}, \quad (3)$$

be the set of splits of \mathcal{N} incompatible with at least one split in $S^{(1)}, \dots, S^{(\ell)}$. We drop the subscript \mathcal{N} when the network is clear from context.

Most split networks cannot be realized as phylogenetic trees. The following is an important special class of more general split networks.

Definition 3 (Circular networks [6]). *A collection of splits \mathcal{S} on \mathcal{X} is called **circular** if there exists a linear ordering (x_1, \dots, x_n) of the elements of \mathcal{X} for \mathcal{S} such that each split $S \in \mathcal{S}$ has the form:*

$$S = \{ \{x_p, \dots, x_q\}, \mathcal{X} - \{x_p, \dots, x_q\} \}$$

for $1 < p \leq q \leq n$. We say that a split network $\mathcal{N} = \{\mathcal{X}, \mathcal{S}, w\}$ is a **circular network** if \mathcal{S} is circular.

Phylogenetic trees, seen as split networks, are special cases of circular networks (e.g. [4]). Circular networks have the appealing feature that they cannot contain too many splits. Indeed, let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a circular network with $|\mathcal{X}| = n$. Then $|\mathcal{S}| = O(n^2)$ [6]. In general, circular networks are harder to interpret than trees are. In fact, they are not meant to represent explicit evolutionary events. However, they admit an appealing visualization in the form of an outer-labeled (i.e., the taxa are on the outside) planar graph that gives some insight into how “close to a tree” the network is. As such, they are popular exploratory analysis tools. We will not describe this visualization and how it is used here, as it is quite involved. See, e.g., [4, Chap. 5] for a formal definition and [8] for examples of applications.

Split networks are naturally associated with a metric. We refer to a function $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, +\infty]$ as a **dissimilarity** over \mathcal{X} if it is symmetric and $d(x, x) = 0$ for all x .

Definition 4 (Metric associated to a split network). *Let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a split network. The dissimilarity $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$ defined as follows*

$$d(x, y) = \sum_{S \in \mathcal{S}_{|x, y}} w(S),$$

for all $x, y \in \mathcal{X}$, is referred to as the metric associated to \mathcal{N} . (It can be shown that d is indeed a metric. In particular, it satisfies the triangle inequality.)

The metric associated with a circular network can be used to reconstruct it.

Definition 5 (*d*-splits). Let $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$ be a dissimilarity. The **isolation index** $\alpha_d(S)$ of a split $S = \{S_1, S_2\}$ over \mathcal{X} is given by

$$\alpha_d(S) = \min\{\tilde{\alpha}_d(x_1, y_1|x_2, y_2) : x_1, y_1 \in S_1, x_2, y_2 \in S_2\},$$

where

$$\tilde{\alpha}_d(x_1, y_1|x_2, y_2) = \frac{1}{2}(\max\{d(x_1, y_1) + d(x_2, y_2), d(x_1, x_2) + d(y_1, y_2), d(x_1, y_2) + d(y_1, x_2)\} - d(x_1, y_1) - d(x_2, y_2)).$$

(Note that the latter is always non-negative.) We say that S is a ***d*-split** if $\alpha_d(S) > 0$.

The following result establishes that circular networks can be reconstructed from their associated metric.

Lemma 1 (*d*-splits and circular networks [6]). Let \mathcal{X} be a set of n taxa and let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a circular network with associated metric d . Then \mathcal{S} coincides with the set of all *d*-splits of $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$. Further the isolation index $\alpha_d(S)$ equals $w(S)$ for all $S \in \mathcal{S}$.

The **split decomposition method** reconstructs $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ from d in polynomial time. When \mathcal{N} is compatible, d is an **additive metric**. See e.g. [2, 5].

In practice one obtains an estimate \hat{d} of d , called the **distance matrix**, from DNA sequences, e.g., through the Jukes-Cantor formula [22] or the log-det distance [23]. The accuracy of this estimate depends on the amount of data used [10, 11]. In previous work in the context of tree reconstruction, distorted metrics were used to encode the fact that large d -values typically produce unreliable \hat{d} -estimates.

Definition 6 (Distorted metrics [13, 14]). Suppose $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ is a split network with associated metric d . Let $\tau, R > 0$. We say that a dissimilarity $\hat{d} : \mathcal{X} \times \mathcal{X} \rightarrow [0, +\infty]$ is a **(τ, R) -distorted metric** of \mathcal{N} if \hat{d} is accurate on “short” distances, that is, for all $x, y \in \mathcal{X}$

$$d(x, y) < R + \tau \quad \text{or} \quad \hat{d}(x, y) < R + \tau \quad \implies \quad |d(x, y) - \hat{d}(x, y)| < \tau.$$

We refer to τ and R as the **tolerance** and **accuracy radius** of \hat{d} respectively.

Distorted metrics have previously been motivated by analyzing Markov models on trees that are commonly used to model the evolution of DNA sequences [10, 11]. Such models have also been extended to split networks [24].

3 Main Results

By the reconstruction result mentioned above, any circular network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ with associated metric d can be reconstructed from a (τ, R) -distorted metric where τ is 0 and R is greater or equal than the diameter $\max\{d(x, y) : x, y \in \mathcal{X}\}$ of \mathcal{N} . In the tree case, it has been shown that a *much smaller* R suffice [10, 11, 14, 17]. Here we establish such results for circular networks.

Chord depth and maximum incompatibility. To bound the tolerance and accuracy radius needed to reconstruct a circular network from a distorted metric, we introduce several structural parameters. The first two parameters generalize naturally from the tree context.

Definition 7 (Minimum weight). Let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a split network. The **minimum weight** of \mathcal{N} is given by

$$\epsilon_{\mathcal{N}} = \min\{w(S) : S \in \mathcal{S}\}.$$

Let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a split network with associated metric d . For a subset of splits $\mathcal{A} \subseteq \mathcal{S}$, we let

$$d(x, y; \mathcal{A}) = \sum_{S \in \mathcal{S}_{|x, y} \cap \mathcal{A}} w(S), \tag{4}$$

be the distance between x and y restricted to those splits in \mathcal{A} .

Definition 8 (Chord depth). Let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a split network with associated metric d . The **chord depth** of a split $S \in \mathcal{S}$ is

$$\Delta_{\mathcal{N}}(S) = \min\{d(x, y; \mathcal{C}_{\mathcal{N}}(S)) : x, y \in \mathcal{X} \text{ such that } S \in \mathcal{S}_{|x, y}\},$$

and the **chord depth** of \mathcal{N} is the largest chord depth among all of its splits

$$\Delta_{\mathcal{N}} = \max\{\Delta_{\mathcal{N}}(S) : S \in \mathcal{S}\}.$$

It was shown in [17, Corollary 1] that, if $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ is compatible, then a (τ, R) -distorted metric with $\tau < \frac{1}{4}\epsilon_{\mathcal{N}}$ and $R > 2\Delta_{\mathcal{N}} + \frac{5}{4}\epsilon_{\mathcal{N}}$ suffice to reconstruct \mathcal{N} in polynomial time (among compatible networks).

For more general circular networks, the minimum weight and chord depth are not sufficient to characterize the tolerance and accuracy radius required for reconstructibility; see Example 1 below. For that purpose, we introduce a new notion that, roughly speaking, measures the extent to which a split network differs from a tree.

Definition 9 (Maximum incompatibility). Let $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ be a split network. The **incompatible weight** of a split $S \in \mathcal{S}$ is

$$\Omega_{\mathcal{N}}(S) = \sum_{S' \in \mathcal{I}(S)} w(S'),$$

and the **maximum incompatibility** of \mathcal{N} is the largest incompatible weight among all of its splits

$$\Omega_{\mathcal{N}} = \max\{\Omega_{\mathcal{N}}(S) : S \in \mathcal{S}\}.$$

We drop the subscript in $\epsilon_{\mathcal{N}}$, $\Delta_{\mathcal{N}}$ and $\Omega_{\mathcal{N}}$ when the \mathcal{N} is clear from context.

Statement of results. We now state our main result.

Theorem 1. *NetworkReconstruction* Suppose $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ is a circular network. Given a (τ, R) -distorted metric with $\tau < \frac{1}{4}\epsilon_{\mathcal{N}}$ and $R > 3\Delta_{\mathcal{N}} + 7\Omega_{\mathcal{N}} + \frac{5}{2}\epsilon_{\mathcal{N}}$, the split set \mathcal{S} can be reconstructed in polynomial time together with weight estimates $\hat{w} : \mathcal{S} \rightarrow (0, +\infty)$ satisfying $|\hat{w}(S) - w(S)| < 2\tau$.

Establishing robustness to noise of circular network reconstruction algorithms is important given that, as explained above, such networks are used in practice to tentatively diagnose deviations from tree-like evolution. Errors due to noise can confound such analyses. See e.g. [8] for a discussion of these issues.

In [17, Sect. 4], it was shown that in the tree case the accuracy radius must depend linearly on the depth. The following example shows that *the accuracy radius must also depend linearly on the maximum incompatibility*.

Example 1 (Depth is insufficient; linear dependence in maximum incompatibility is needed). Consider the two circular networks in Fig. 1. In both networks, $\mathcal{X} = \{x_1, x_2, y_1, y_2\} \cup \{z_0, z_1, \dots, z_n\}$, and the n vertical lines, the horizontal line, and the two arcs are splits of weight 1. The chord depth of both networks is 1 while their maximum incompatibility is n . In both networks

- $d(z_i, x_j) = i + 1, 0 \leq i \leq n, 1 \leq j \leq 2,$
- $d(z_i, y_j) = n - i + 1, 0 \leq i \leq n, 1 \leq j \leq 2,$
- $d(x_1, x_2) = d(y_1, y_2) = 2,$
- $d(x_1, y_2) = d(x_2, y_1) = n + 2.$

The only difference is that, in graph (A), $d(x_1, y_1) = n + 2$ and $d(x_2, y_2) = n$ while, in graph (B), $d(x_2, y_2) = n + 2$ and $d(x_1, y_1) = n$. If we choose the distance matrix \hat{d} as follows:

- $\hat{d}(x_1, y_1) = \hat{d}(x_2, y_2) = n + 1,$
- $\hat{d} = d$ for all other pairs,

then \hat{d} is a $(\tau, n - 1)$ -distorted metric of both networks for any $\tau \in (0, 1)$. Hence, these two circular networks are indistinguishable from \hat{d} . Observe that the chord depth is 1 for any n , but the maximum incompatibility can be made arbitrary large. (Note that the claim still holds if we replace the chord depth with the “full chord depth” $\max\{\min\{d(x, y) : x, y \in \mathcal{X}, S \in \mathcal{S}|_{x,y} : S \in \mathcal{S}\},$ which also includes weights of incompatible splits separating x and y .)

Proof idea. Our proof of Theorem 1 is based on a divide-and-conquer approach of [17], first introduced in [14] and also related to the seminal work of [10, 11] on short quartet methods and the decomposition methods of [19, 20]. More specifically, we first reconstruct sub-networks in regions of small diameter. We then extend the bipartitions to the full taxon set by hopping back from each taxon to this small region and recording which side of the split is reached first. However, the work of [17] relies heavily on the tree structure, which simplifies many arguments. Our novel contributions here are twofold:

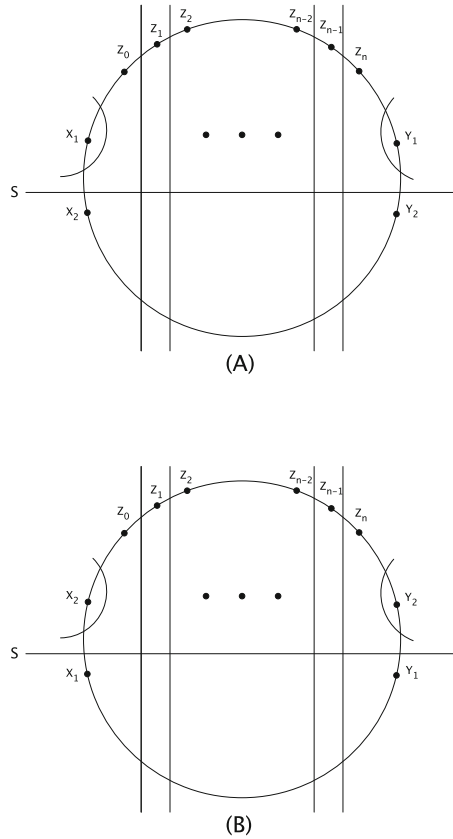


Fig. 1. Two circular networks indistinguishable from a distorted metric with sublinear dependence on the maximum incompatibility. Here the taxa are ordered on a circle and lines indicate splits. For instance, in (A), the leftmost vertical line is the split with $\{z_0, x_1, x_2\}$ on one side and all other taxa on the other. In both networks, $\mathcal{X} = \{x_1, x_2, y_1, y_2\} \cup \{z_0, z_1, \dots, z_n\}$, and the n vertical lines, the horizontal line, and the two arcs are splits of weight 1.

- We define the notion of maximum incompatibility and highlight its key role in the reconstruction of circular networks, as we discussed above.
- We extend the effective divide-and-conquer methodology developed in [10, 11, 14, 17, 19, 20] to circular networks. The analysis of this more general class of split networks is more involved than the tree case. In particular, we introduce the notion of a compatible chain—an analogue of paths in graphs—which may be of independent interest in the study of split networks.

Details are provided in [25].

Acknowledgements. Work supported by NSF grants DMS-1007144, DMS-1149312 (CAREER), DMS-1614242 and CCF-1740707 (TRIPODS).

References

1. Felsenstein, J.: *Inferring Phylogenies*. Sinauer, Sunderland (2004)
2. Steel, M.: *Phylogeny—Discrete and Random Processes in Evolution*. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 89. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2016)
3. Warnow, T.: *Computational Phylogenetics: An Introduction to Designing Methods for Phylogeny Estimation*. Cambridge University Press, Cambridge (2017)
4. Huson, D.H., Rupp, R., Scornavacca, C.: *Phylogenetic Networks: Concepts Algorithms and Applications*. Cambridge University Press, Cambridge (2010)
5. Semple, C., Steel, M.: *Phylogenetics*. Mathematics and its Applications Series, vol. 22. Oxford University Press, Oxford (2003)
6. Bandelt, H.J., Dress, A.W.M.: A canonical decomposition theory for metrics on a finite set. *Adv. Math.* **92**(1), 47–105 (1992)
7. Nakhleh, L., Morrison, D.: Phylogenetic networks. In: Kliman, R.M. (ed.) *Encyclopedia of Evolutionary Biology*, pp. 264–269. Academic Press, Oxford (2016)
8. Huson, D.H., Bryant, D.: Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol.* **23**(2), 254–267 (2006)
9. Bryant, D., Moulton, V.: Neighbor-Net: an agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol.* **21**(2), 255–265 (2004)
10. 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. Algorithms* **14**(2), 153–184 (1999)
11. 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)
12. 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)
13. King, V., Zhang, L., Zhou, Y.: On the complexity of distance-based evolutionary tree reconstruction. In: *2003 Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 444–453. SIAM, Philadelphia (2003)
14. Mossel, E.: Distorted metrics on trees and phylogenetic forests. *IEEE/ACM Trans. Comput. Bio. Bioinform.* **4**(1), 108–116 (2007)
15. Cryan, M., Goldberg, L.A., Goldberg, P.W.: Evolutionary trees can be learned in polynomial time. *SIAM J. Comput.* **31**(2), 375–397 (2002)
16. Mossel, E., Roch, S.: Learning nonsingular phylogenies and hidden Markov models. *Ann. Appl. Probab.* **16**(2), 583–614 (2006)
17. Daskalakis, C., Mossel, E., Roch, S.: Phylogenies without branch bounds: contracting the short, pruning the deep. *SIAM J. Discrete Math.* **25**(2), 872–893 (2011)
18. Gronau, I., Moran, S., Snir, S.: Fast and reliable reconstruction of phylogenetic trees with indistinguishable edges. *Random Struct. Algorithms* **40**(3), 350–384 (2012)
19. Huson, D.H., Nettles, S.M., Warnow, T.J.: Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *J. Comput. Biol.* **6**(3–4), 369–386 (1999)
20. Roshan, U.W., Moret, B.M.E., Warnow, T., Williams, T.L.: Rec-I-DCM3: a fast algorithmic technique for reconstructing large phylogenetic trees. In: *International Computational Systems Bioinformatics Conference*, pp. 98–109. IEEE Computer Society (2004)
21. Buneman, P.: The recovery of trees from measures of dissimilarity. In: Kendall, D.G., Tautu, P. (eds.) *Mathematics in the Archaeological and Historical Sciences*, pp. 387–395 (1971)

22. Jukes, T.H., Cantor, C.R.: Evolution of protein molecules. In: *Mammalian Protein Metabolism*, pp. 21–132. Academic Press, New York (1969)
23. Steel, M.: Recovering a tree from the leaf colourations it generates under a Markov model. *Appl. Math. Lett.* **7**(2), 19–23 (1994)
24. Bryant, D.: Extending tree models to split networks. In: Pachter, L., Sturmfels, B. (eds.) *Algebraic Statistics for Computational Biology*, pp. 297–310. Cambridge University Press, Cambridge (2005)
25. Roch, S., Wang, K.C.: Circular networks from distorted metrics. Preprint (2017). [arXiv:1707.05722](https://arxiv.org/abs/1707.05722)



A Nested 2-Level Cross-Validation Ensemble Learning Pipeline Suggests a Negative Pressure Against Crosstalk snoRNA-mRNA Interactions in *Saccharomyces Cerevisiae*

Antoine Soulé^{1,2}, Jean-Marc Steyaert², and Jérôme Waldispühl¹

¹ School of Computer Science, McGill University, Montréal, Canada
jeromew@cs.mcgill.ca

² LIX - UMR 7161, École Polytechnique, Palaiseau, France

Abstract. The growing number of RNA-mediated regulation mechanisms identified in the last decades suggests a widespread impact of RNA-RNA interactions. The efficiency of the regulation relies on highly specific and coordinated interactions, while simultaneously repressing the formation of opportunistic complexes. However, the analysis of RNA interactomes is highly challenging due to the large number of potential partners, discrepancy of the size of RNA families, and the inherent noise in interaction predictions.

We designed a recursive 2-step cross-validation pipeline to capture the specificity of ncRNA-mRNA interactomes. Our method has been designed to detect significant loss or gain of specificity between ncRNA-mRNA interaction profiles. Applied to snoRNA-mRNA in *Saccharomyces Cerevisiae*, our results suggest the existence of a repression of ncRNA affinities with mRNAs, and thus the existence of an evolutionary pressure inhibiting such interactions.

Keywords: RNA · RNA-RNA interaction · Ensemble learning

1 Introduction

Evidence of the breadth of the role of ribonucleic acids in gene regulation are now multiplying. For instance, in eukaryotes microRNAs bind mRNAs to control gene expression [1], and in prokaryotes the OxyS RNA interacts with the *hflA* mRNA to prevent ribosome binding and thus inhibit translation [2].

Among all non-coding RNAs (ncRNAs) already identified, the category of small nucleolar RNAs (snoRNAs) is of particular interest. snoRNAs form a large class of well-conserved small ncRNAs that are primarily associated with chemical modifications in ribosomal RNAs (rRNAs) [3]. Recent studies revealed that orphan snoRNAs can also target messenger RNAs (mRNAs) in humans [4] and

mice [5], and probably contribute to regulate expression levels. However, despite recent investigations there are to date no evidence that similar snoRNA-mRNA interactions occur in simpler unicellular microorganisms [6,7].

Interestingly, it turns out that RNA-based gene regulation mechanisms have been primarily linked to higher eukaryotes [8], although it is still not clear if this observation results from an incomplete view of RNA functional landscape or the existence of a negative pressure preventing RNA to interfere with other transcripts.

Our understanding of RNA-mediated regulation mechanisms significantly improved in recent years. In addition to well-documented molecular pathways (e.g. [2]), regulation can also occur at a higher level through global affinities between ncRNAs and mRNAs populations [9]. Furthermore, Umu *et al.* [10] showed another intriguing, yet complementary, level of control of gene expression that could explain discrepancies previously observed between expressions of mRNAs and the corresponding protein expressions in bacteria [11,12]. In their study, the researchers extracted a signal suggesting a negative evolutionary pressure against random interactions between ncRNAs and mRNAs that could reduce translation efficiency. However, these results cannot be trivially extended to eukaryotes where the role of the nucleus has to be considered.

In this study, we investigate this phenomenon of avoidance of random interactions between ncRNA and mRNA in *Saccharomyces Cerevisiae*. In particular, we focus our analysis on the bipartite interactome between snoRNAs and mRNAs. Indeed, the snoRNA family is an ancient and large class of ncRNAs for which the mechanism of mRNA avoidance could explain the absence of known interactions between snoRNAs and mRNAs in unicellular eukaryotes.

A major challenge of this analysis stems from severely unbalanced datasets. While we retrieve more than 6000 annotated mRNAs, we could only recover less than one hundred snoRNAs [13]. Such disparity is a serious source of bias that should be carefully addressed. Therefore, we developed a customized ensemble learning pipeline to quantify the specificity of RNA binding profile between unbalanced RNA families.

First, we use state-of-the-art prediction tools to compute the snoRNA-mRNA interactome as the set of all interactions between snoRNAs and mRNAs. Then, we design an ensemble learning pipeline to identify statistically significant biases in the distribution of binding affinities between classes of RNAs. Importantly, in order to remove any possible source of bias during the parametrization of classifiers, we introduce a second level of Leave-One-Out Cross-Validation (LOOCV) to avoid overfitting. Our results reveal that although classes of snoRNAs exhibit preferential interaction patterns with mRNAs, this selective pressure is not as strong as initially anticipated. It corroborates previous hypothesis on prokaryotes, and suggests the presence of a phenomenon of avoidance of random interactions between ncRNAs and mRNAs in single-celled eukaryotes.

2 Approach

We aim to characterize the strength and specificity of random ncRNA-mRNA interactions in *Saccharomyces Cerevisiae*, although our work primarily focuses on snoRNA-mRNA interactions. Our data set includes smaller categories of ncRNAs (e.g. spliceosomal RNAs) used for an additional control of our results.

We computed ncRNA-mRNA interactomes from ncRNAs and mRNAs sequences using two different state-of-the-art computational prediction tools (RNAup [14] and intaRNA [15]). Those predictions are to serve as an approximation for the propensity of those ncRNAs to form crosstalk interactions with mRNAs. By using an ensemble learning pipeline, we approximated the specificities of interaction profiles in those interactomes. We also approximated the specificities of ncRNAs sequences with machine learning upon the Kmer compositions of the said sequences. The comparison of the approximated specificities highlights a global pressure inhibiting the affinity ncRNA-mRNA interactions in *Saccharomyces Cerevisiae*.

We finally completed this work by a collection of complementary control tests providing a better understanding of the limitations of this work. Data, code, raw results and supplementary displays are available at http://jwgitlab.cs.mcgill.ca/Antoine/nested_loocv_pipeline/tree/master.

3 Methods

3.1 Dataset

Saccharomyces Cerevisiae. We focus our study on a single organism: *Saccharomyces Cerevisiae*. Working on a single organism ensures that all the molecules co-evolved and that their interactions were under the same evolutionary pressure. We also focus our study on a eukaryote to investigate the influence of the nucleus. Indeed, the nuclear membrane creates a confined environment that segregates molecules. Moreover, eukaryotes usually display more complex mechanisms and have more coding sequences than prokaryotes and archaea. Extending the study to a family instead or even further, like Umu *et al.* [10] did for instance, has been considered. However, less data are available for other related yeasts and including more species increases the number of parameters to consider. We came to the conclusion that a multi-species study, while being interesting, was unrealistic yet. Finally, we excluded multicellular organisms to avoid problematic phenomena like specialized tissues.

For all those reasons, this study required a unicellular eukaryote offering a satisfying number of identified RNA sequences and *Saccharomyces Cerevisiae* appeared to be the most suited model by being a model eukaryote organism with the greatest number of annotated sequences amongst unicellular eukaryotes.

All sequences have been obtained from the manually curated Genolevure [13] database.

ncRNA-mRNA Interactome. Our main source of features is the ncRNA-mRNA interactome i.e. all the ncRNA-mRNA interactions. Noticeably, we are referring to ncRNA-mRNA interactions as computational predictions instead of experimentally observed interactions. The probability of such event is conventionally approximated by the energy barrier and difference of entropy (Δg) between the structures of the two molecules and the structure of a potential complex [14]. We work under the usual and reasonable assumption that, for two complexes i and j , if $\Delta g_i < \Delta g_j$ then the complex i is more stable than the complex j and thus is more likely to form and be observed. In order to study the set of all potential ncRNA-mRNA complexes, we computed for all {ncRNA,mRNA} pairs the corresponding Δg using prediction tools (cf. Sect. 3.2), thus resulting in two predicted ncRNA-mRNA interactomes: one for each prediction tool we used (See Sect. 3.2).

We focused our study on ncRNA-mRNA interactome for two reasons. First, the role of mRNA as temporary medium of genetic material makes it a central element in most cellular pathways. mRNAs are centrepieces of several mechanisms such as regulation [10–12] and splicing that might be impacted by crosstalk interactions. Second, ncRNAs (i.e. non-coding RNA, which refers here to RNA which are neither messenger, transfer or ribosomal RNA and also excludes miRNA and siRNA cf. Sect. 3.1) offer properties of interest for this study.

Indeed, the selected ncRNAs can be clustered into categories sharing similar properties, such as structure and length, which makes any comparison more meaningful. Those ncRNAs are also free from cellular mechanisms such as maturation or directed export that might generate noise. Finally, there is no observed interaction between those ncRNAs and mRNAs. As a consequence we can assume that the interaction we predict are opportunistic and not part of a defined biological pathway. A detailed description of ncRNAs labels is provided in Sect. 3.1 and in the supplementary material.

We also considered two other practical aspects in this decision: maximizing the number of available annotated sequences and maximizing the number of crosstalk interactions (i.e. minimizing the number of known interactions). The first aspect directly impacts the statistical validity of any potential results and the second is justified by the goal of this study. The ncRNA-mRNA interactome also satisfies those two aspects.

ncRNA Labels. In order to conduct this study, we had to choose which of the mRNAs or ncRNA to label. The absence of structural properties in mRNA naturally inclined us to label ncRNA instead. We produced a 5-label classification (cf. Table 1) according to the gene ontologies based on both functional and structural properties. Out of those five labels, two labels happen to be much more similar in terms of lengths and numbers. As a consequence, we performed all our tests with both the five labels dataset and a dataset limited to those two similar labels and are providing displays for both.

Table 1. Numbers for both 5-label and 2-label datasets, means and standard deviations of distributions of sequence lengths and the colours associated in our displays for each ncRNA label

ncRNA Label	Dataset		Length		colour
	5 labels	2 labels	μ	σ	
miscellaneous	11	0	900.27	828.04	black
C/D box	45	45	106.87	34.41	red
H/ACA box	29	29	270.83	176.75	blue
spliceosomal	5	0	245.60	183.40	yellow
unknown	7	0	500.14	225.31	green
total	97	74	281.39	383.93	

A complete description of all those ncRNA is available in the supplementary material. For the sake of clarity, we will only provide a shorter description of each label in this paper.

C/D box and H/ACA box ncRNAs are snoRNAs (small nucleolar RNA) involved in pre-rRNA maturation by performing two different modifications of specific bases. C/D box snRNAs are performing pseudouridylation, an isomerization of uridines into pseudouridines. Pseudouridines have an extra NH group able to form supplementary hydrogen bonds. Those bonds stabilize rRNA structure [16, p. 200]. H/ACA box snRNAs are performing 2-O methylation, a methylation of the ribose. RNA has a short lifespan compared to DNA. By methylating the ribose, the rRNA is less vulnerable to degradation by bases or RNAses. In addition to this increased lifespan, this modification also impacts the rRNA structure by changing spatial constraints and decreasing the number of hydrogen bonds the modified base can form [16, p. 200].

Those two labels are the most consistent, both in numbers and internal similarities with both sequential constraints (boxes) and similar structures common to all the ncRNAs of a given label. Moreover, the lengths of ncRNAs are consistent inside each label and shorter than the ncRNA average (cf. Fig. 4).

Importantly, we will use these two groups (i.e. C/D box and H/ACA box ncRNAs) to study the existence of an evolutionary pressure on snoRNAs. The other groups described below will be used as control and/or to suggest the generalization of the pressure to other classes of ncRNAs.

Spliceosomal ncRNAs share the common trait of being involved in the splicing process. However all other properties vary.

Miscellaneous ncRNAs have been identified and their functions are known. However those functions are too specific and diverse to be gathered in any label but Miscellaneous. Moreover all other properties vary.

Unknown ncRNAs have been identified but, unlike miscellaneous ncRNAs, their functions remain unknown. Moreover all other properties vary in an even wider range than the two previous labels.

3.2 Features Description

This section describes the three metrics used in this study to produce the main sets of features: RNAup, IntaRNA and Kmer composition similarity. Other basic features used as control, such as sequence length, are not described as they are straightforward.

RNA-RNA Interactions Prediction Tools. In order to produce a satisfying interactome we use two different RNA-RNA interaction prediction tools: RNAup [14] and IntaRNA [15, 17]. We selected nonspecialized prediction tools over specialized ones such as RNAsnoop [18] as we are interested in non-specific interaction.

Both RNAup and IntaRNA implement the same core strategy. They compute the hybridization energies between the two RNAs as well as the accessibility (i.e. probability of being unpaired) for each interaction site. Those values are then combined to score potential interaction sites. The highest scoring sites are returned together with the free energy of binding. We can then retrieve the secondary structures of each individual RNA using constraint folding algorithms.

RNAup strictly implements this strategy thus predicting the optimal minimum free energy (MFE) compatible with the axioms. IntaRNA differs by two aspects. The first one is that the version of IntaRNA used in this work uses a slightly less recent version of Turner energies model. However the differences between those versions are minor and are very unlikely to produce the observed dissimilarities. The second one is that IntaRNA adds a seeding step to reject interaction sites deemed unlikely. This extra step reduces the search space by focusing on the most promising ones and significantly reduces the runtimes compared to RNAup. An extensive description of the seeding procedure is presented by *Bush et al.* [17]. Comparative benchmarks place IntaRNA in the top of prediction tools with better scores than RNAup [19, 20]. Indeed, IntaRNA appears to predict interactions closer to the observed ones compared to predictions from others prediction tools, including RNAup. As a consequence this heuristic seems well founded and efficient.

In this study we used this difference between RNAup and IntaRNA to predict two slightly different interactions modes. For each {ncRNA,mRNA} pair, we are assuming that RNAup outputs the optimal MFE regardless of its likelihood while IntaRNA outputs a probably weaker but more realistic interaction. Since realistic interactions are more likely to be observed in the cell than the theoretical optimums, any pressure should impact the first before the second. As a consequence, we aimed at highlighting such pressure by studying those two sets of interactions in parallel.

Kmer Composition Similarity. In addition to the two prediction tools mentioned in the previous subsection, we use a third metric: the similarity of the

Kmer composition of ncRNA sequences. The term *Kmer* refers here to every possible sequence of nucleobasis of length K . This metric associates to each ncRNA the distribution of each Kmer in its sequence, including repetitions. The set of all those distributions is gathered as a vector space suitable for machine learning (cf. Sect. 3.3). We produced this third set of features in order to assess the specificity of the sequence and to provide a reference point to the two other sets of features.

All experiments involving Kmers have been made with $K = 5$ for two reasons. The first one is that five is the length of the average interacting zone in RNA-RNA interactions and so is a suitable length to capture any key subsequences impacting those interactions. The second one is that the number of Kmers to consider grows with the value of K . $K = 5$ offers the advantage of being both manageable in term of cost and also results in a number of dimensions comparable to the two other methods (i.e. RNAup and IntaRNA). We performed preliminary tests with others values, especially $K = 6$. Those tests showed little to no differences.

3.3 Ensemble Learning Pipeline

The overall goal of our machine learning approach is to investigate a possible bias affecting ncRNA-mRNA crosstalk interactions. In order to do so, we compare the specificity of ncRNA sequences with the specificity of ncRNA-mRNA interaction profiles. The specificity of ncRNA sequences is approximated by the ability of classifiers to predict the labels of ncRNAs from their Kmer composition. ncRNA-mRNA interaction profiles are predicted using prediction tools and their specificity is approximated by the ability of classifiers to predict the labels of ncRNAs from those profiles (Fig. 1).

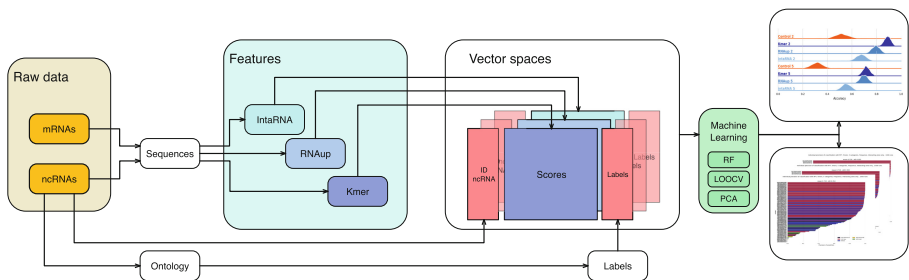


Fig. 1. Illustration of our ensemble learning pipeline. The process starts from RNAs data in orange. Each ncRNA will be associated to a vector in the vector spaces and will be attributed a label according to its ontology. From either ncRNAs sequences (Kmer) or both ncRNA and mRNAs sequences (IntaRNA, RNAup), a set of scores will be computed and used as features. Machine learning is finally used to produce the results we are presenting from those vector spaces.

Our utilization of machine learning in this project is challenging for two different reasons that justify all the following methodology choices:

1. The ratios $|vectors|/|features|$ of our datasets are problematic: 97 vectors for 6663 dimensions for the vector spaces built from the interactomes. Those ratios are due to both cellular biology, since the number of mRNAs in a genome is always greater by several folds to the number of ncRNAs, and the limited availability of annotated ncRNAs sequences thus limiting as well the number of vectors. Those two issues are beyond our control and, to our knowledge, there is no way for us to significantly improve those ratios for *Saccharomyces Cerevisiae* without considerable drawbacks. Moreover *Saccharomyces Cerevisiae* already has the greater number of annotated ncRNAs amongst similar organisms.
2. Our goal is neither to train a good classifier nor to classify unlabelled RNAs but to estimate how well the labels can be predicted from the different features. We are working under the reasonable assumption that a loss in performance between two sets of features implies that the lesser performing set is less specific. If the two sets of features are related, like ours are, it would imply a levelling mechanism.

Leave-one-out Cross-validation (LOOCV). Cross-validation refers in machine learning to partitioning the data set into different sets to separate the data used to train the classifier and the ones used to test it. The goal of cross-validation is to ensure the credibility of the results produced.

We use a leave-one-out cross-validation technique (LOOCV) for validation. For every vector v_i in our set V of vectors we train a classifier on the set $(V - v_i)$ and test the resulting classifier on the vector v_i . The final accuracy is computed as the average of the accuracies for all vectors. This technique fits our data set and its limited number of vectors. A more classical approach such as train-validation-test would have required us to use very small sets.

Importantly, we are also performing a second nested level of LOOCV to avoid any bias during the parametrization of the classifiers. This second level is described in Sect. 3.3 and illustrated in *Algorithm 1*.

Principal Component Analysis (PCA). Since the ratio $|vectors|/|features|$ is poor in the dataset, it may hinder the accuracy of the classifiers. Principal component analysis (PCA) is a standard method to improve this ratio by reducing the number of dimensions. The PCA uses an orthogonal transformation to build a set of uncorrelated features (components) from the initial features with the objective of maximizing variance (i.e. minimizing the information lost by transforming).

Algorithm 1: 2-level_procedure(Df, l_nc)

Data: Given dataframe D (i.e., the vector space) and l_nc , the list of numbers of components to consider for the PCA

Result: Returns $d_accuracy$, dictionary of (vector, accuracy)

```

dict d_accuracy =  $\emptyset$ ;
for vector  $v_i \in D$  do
  test_set =  $\{v_i\}$ 
  train_set =  $D - \{v_i\}$ 
  best_nc = -1
  accuracy_best_nc = -1
  for  $nc \in l\_nc$  do
     $D' = D - \{v_i\}$ 
     $D' = PCA(D', nc)$ 
    for vector  $v_j \in D'$  do
      sub_test_set =  $\{v_j\}$ 
      sub_train_set =  $D' - \{v_j\}$ 
      classifier = new RFT_classifier()
      classifier.train(sub_train_set)
      tmp_accuracy = classifier.test(sub_test_set)
      if tmp_accuracy > accuracy_best_nc then
        best_nc = nc
        accuracy_best_nc = tmp_accuracy
    end for
  end for
   $D = PCA(D, best\_nc)$ 
  classifier = new RFT_classifier()
  classifier.train(train_set)
   $d\_accuracy[v_i] = classifier.test(test\_set)$ 
end for
return d_accuracy;

```

The number of components to transform to is an important parameter that may influence the classifier accuracy. Performing preliminary tests to determine the best number would lead to a serious risk of overfitting. As a consequence we dynamically determined this number for each vector. The procedure is described in Algorithm 1. From the first LOOCV, the set of vectors V has been split into a set of pairs of a training set $V' = (V - \{v_i\})$ and a test set $\{v_i\}$. For each pair, a second LOOCV is performed on V' leading to another set of pairs of a training set $V'' = (V' - \{v_j\}) = (V - \{v_i, v_j\})$ and a test set $\{v_j\}$. Potential values for the number of components are tested and the one producing the best accuracy over V' is selected and used on V to predict the label of v_i . As a consequence, the number of components to transform to is always selected independently from the test set.

Ideally, the set of potential values for the number of components would be $1, 2, \dots, |V|$. However the computation time grows linearly with the number of values tested. As a consequence we decided to use a subset of $1, 2, \dots, |V|$ instead. Preliminary tests shows a light peak of performances at 8–10 components with a slight decrease before and after. As a consequence we tried all values from $1, 2, \dots, 20$. We also added 0 (i.e. not performing a PCA).

Random Forest (RF) Classifier. We chose to use ensemble learning and more specifically Random Forest (RF) classifiers over other methods and classifiers because of some anticipated properties of the datasets. Indeed, the limitations of prediction tools are likely to generate noise which RF are relatively resilient to [21, p.596]. Moreover, the interactions we aimed at capturing were likely to be complex and the size of the training set to be limited. Since RF can capture complex interactions and are simple to train [21, p. 587] compared to other classifiers [21, p. 587] they appeared to be a fitting candidate.

Our implementation uses the python package Scikit-learn [22].

As the name suggests, Random Forest classifiers involve randomness. As a consequence we repeated the procedure and display distributions in order to counterbalance the variation of the predictions. Preliminary results show that the average accuracies of those distributions converge (10^{-4}) within the first 500 runs. However we decided to double this value to add a comfortable security margin.

Dummy Classifier. A second classifier is trained in parallel to serve as a control. As the name “dummy” suggests, it is not an actual classifier but an heuristic randomly generating labels for the test set according to the probabilities distribution it extracted from the training set. As the dummy classifier is always trained and tested on the same sets as its RFT counterparts, it appears to be a suitable solution to produce a sound control while using LOOCV and using unbalanced labels. However, as all dummy classifiers produced extremely close performances, we decided to display only one of the dummy classifiers in each display instead of one per other classifier for the sake of clarity. Please note that the dummy classifier is unaffected by PCA as it does not consider the features.

Performance Metric for a Multi-label Dataset. The number of labels in our data sets prevents straightforward use of some classical displays such as ROC curves. A single prediction can indeed be, for instance and at the same time, both a false positive for a given label and a true negative for another. As a consequence we have $TPR + FPR + TNR + FNR \geq 1$ ([True, False] [Positive, Negative] Rate) and plotting one ROC curve for each label offers little readability. As a consequence we instead chose to use displays based on accuracy ($Accuracy = Precision = |True Predictions|/|Predictions|$).

3.4 Main Experiments

As described in Sects. 3.2 and 3.1, our work associates each ncRNA with three vectors of features ($\{Kmer, \text{intaRNA}, \text{RNAup}\}$) and a label. By doing so we produced three vector spaces which are suitable for machine learning. We consider that the ability of the classifiers to predict those labels reflects the said specificity. Therefore, the goal of the machine learning procedure described in Sect. 3.3 is to assess the specificity of the sets of features regarding the labels.

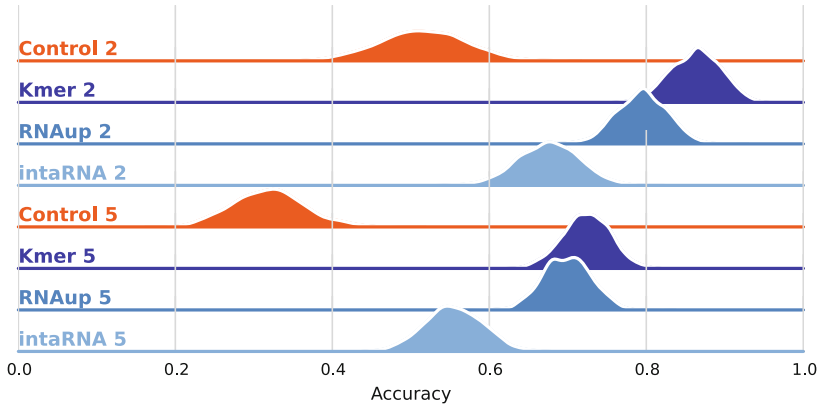


Fig. 2. Distribution of accuracies of 1000 classifiers following specifications of Sect. 3.3. Each row corresponds to either a set of features ($\{\text{Kmer, RNAup, intaRNA}\}$, cf. Sect. 3.2) or the control (cf. Sect. 3.3) associated with a number of labels ($\{2, 5\}$, cf. Sect. 3.1). Means and standard deviations for all distributions are displayed in Table 2. (Color figure online)

Table 2. Means (μ) and standard deviations (σ) for all distributions displayed in Fig. 2

	2 labels		5 labels	
	μ	σ	μ	σ
Control	0.517	0.057	0.316	0.044
Kmer	0.863	0.031	0.724	0.03
RNAup	0.794	0.032	0.699	0.03
intaRNA	0.675	0.04	0.555	0.037

Figure 2 displays the distribution of the accuracies of the classifiers for all three kinds of features with two or five labels. The combination of LOOCV (cf. Sect. 3.3) and the inherent randomness of RF classifiers (cf. Sect. 3.3) lead us to produce and display distributions of accuracies instead of a single value. Exact means (μ) and standard deviations (σ) values for all those distributions are displayed aside in Table 2. The best accuracies are obtained from the Kmer similarity scores with 86.3% of correct prediction with two labels and a standard variation of only 0.31%. Results obtained from scores predicted by RNAup are less accurate but are still very distinct from the control with no overlapping. However, results obtained from scores predicted by IntaRNA are significantly less accurate to the point that the distribution overlaps with the control. Results obtained with five labels display a similar hierarchy between Kmer, RNAup and IntaRNA with the addition of an expected global loss of accuracy. Indeed, the increased number of labels to predict makes the problem harder as shows the important drop of accuracy of the control. However, Kmer, RNAup and IntaRNA appear to all be more resilient than the control to this change.

This first display suggests that the interaction profiles predicted by IntaRNA are significantly less specific than the ones predicted by RNAup. The interaction profiles predicted by RNAup also appear to be the closest to the ones produced from Kmer similarity scores and thus seem to give the most accurate account of the specificities of the sequences. This observation together with the difference between the two prediction tools described in Sect. 3.2 suggest that probable interactions (i.e. the ones predicted by IntaRNA) are more inhibited than the potential optimal ones (i.e. the ones predicted by RNAup). This first observation is coherent with the influence of an evolutionary pressure as the inhibition of probable interactions would have a greater impact than the inhibition of potential optimal ones which are less likely to form.

Figure 3 is a different presentation of the results displayed in Fig. 2. Raw results from the classifiers are unitary predictions (i.e. predictions of the label of one vector). We gathered those unitary predictions for each vector, thus producing an averaged accuracy for each of them. Figure 3 aims at highlighting variations inside the distribution displayed in Fig. 2. Please note that each column corresponds now to a different set of features while the upper row displays the results with two labels and the lower row displays the results with five labels. Each line corresponds to a ncRNA, the length reflecting the accuracy of predictions made for this ncRNA label while the colour corresponds to its label. Please also note that lines are sorted by accuracies. As a consequence, the order varies in all of those six subgraphs.

The drop of accuracy observed in Fig. 2 between Kmer similarity scores, RNAup predicted scores and IntaRNA scores is also visible in Fig. 3 as a more concave slope for better performing sets of features. However Fig. 3 also displays variations of accuracies from one label to the other. C/D box RNAs (red) are the most noticeable group as those RNAs are, on average, extremely well-predicted with all features and either two and five labels. H/ACA box RNAs (blue), on the other hand, seem to be harder to predict from Kmer similarity scores or RNAup predicted scores than C/D box RNAs but show a dramatic drop of accuracy in predictions made from IntaRNA predicted scores. Predictions accuracies of the three remaining labels vary from a set of features to the other and even inside a label for a given set of features. We have been unable so far to determine if this was only due to a lesser number of vectors for those labels or to other parameters.

Results displayed in Fig. 3 complement our previous observations as the {Kmer, RNAup, IntaRNA} hierarchy is still clearly observable. However, Fig. 3 displays a phenomenon invisible in Fig. 2: the variance in predictions accuracy between the label, especially regarding C/D box RNAs and H/ACA box RNAs. Indeed, predictions for C/D box RNAs (red) are always the most accurate while predictions for H/ACA box RNAs (blue) clearly fall behind. This variance goes from a limited difference (most predictions for H/ACA box RNAs are still above 80% accuracy in predictions from Kmer similarity scores with two labels, cf. top left graph) to a dramatic drop (predictions from IntaRNA predicted scores with two labels, cf. top right graph). Predictions accuracies of the three remaining

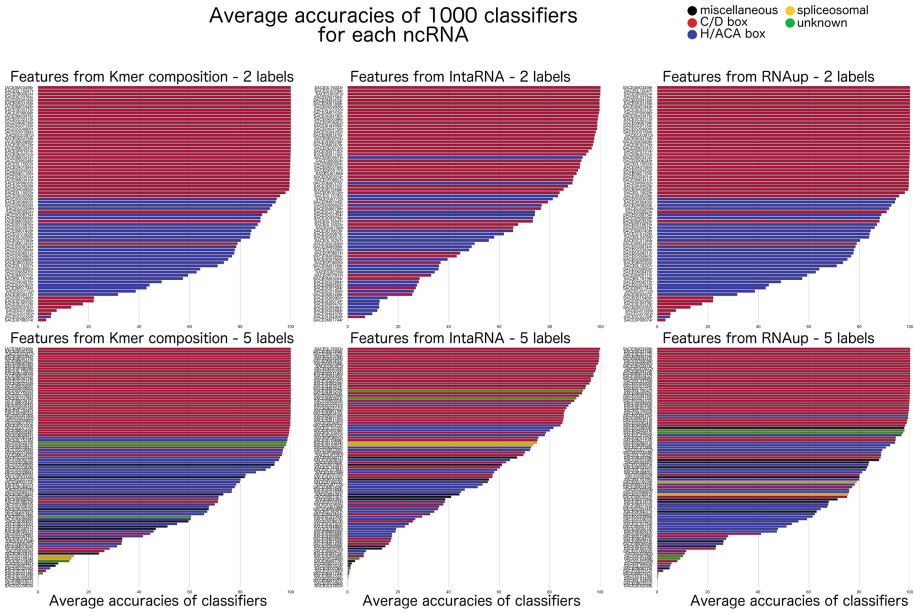


Fig. 3. Average accuracies of classifiers for each vector (i.e. ncRNA) over 1000 tests. Each column corresponds to a set of features ($\{Kmer, RNAUp, intaRNA\}$, cf. Sect. 3.2). The first row displays the results with five labels and the second with two labels (cf. Sect. 3.1). Each colored line in those 6 displays corresponds to a [vector/ncRNA]. The length of the line represents the averaged accuracy of the 1000 classifiers for the corresponding [vector/ncRNA]. The colour of the line corresponds to the label of the associated [vector/ncRNA]. Please note that [vectors/ncRNAs] are sorted according to the accuracy associated to them. As a consequence the order is different in all six graphs. (Color figure online)

labels vary from a set of features to the other and even inside a label for a given set of features. We have been unable so far to determine if this was only due to a lesser number of vectors for those labels or to other parameters. However, the predictions of the three remaining labels display accuracies similar to the ones of predictions for H/ACA box RNAs. Since the dataset contains more H/ACA box RNAs than the three other labels put together, this similarity stresses that H/ACA box RNAs are way harder to predict than C/D box RNAs. Further discussions of this difference of performances between labels require to first introduce Fig. 4.

In order to investigate the drop in accuracy between predictions made from scores predicted by RNAUp and IntaRNA we plotted distributions of scores as box plots for each tool (left and middle) and for each ncRNA labels (colours). We also plotted the distributions of the lengths of ncRNAs sequences (right, please note that the influence of length results is discussed in Sect. 3.5). The results are displayed in Fig. 4. The colour code is the same as in Figs. 2 and 3. Figure 4 shows

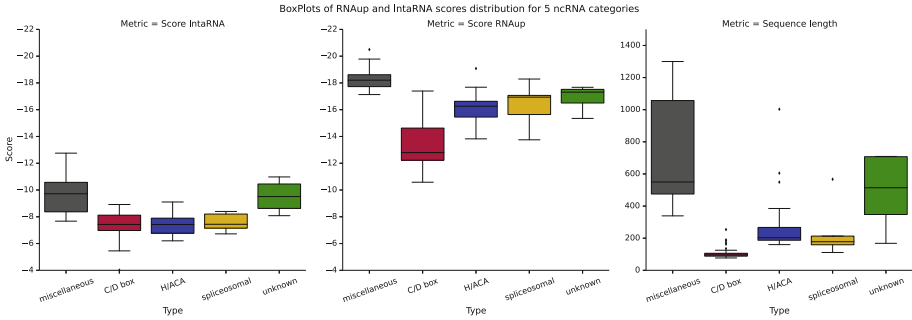


Fig. 4. Normalized distributions of scores predicted by IntaRNA (left), scores predicted by RNAup (middle) and lengths of sequences (right) for each ncRNA label. Scores are in kcal/mol. Please note that scores from both IntaRNA and RNAup approximate a difference of entropy (Δg) and are therefore negative. A lower score thus suggests that the interaction is stronger. (Color figure online)

that RNAup is not only outputting stronger scores (entropy scores are negative cf. Sect. 3.1) but also preserves distinctions between the labels, especially between C/D box RNAs and H/ACA RNAs scores. This observation is coherent with the better performances of classifiers learning from the interactome predicted with RNAup. However the important drop in accuracies displayed in Fig. 3 on scores predicted with IntaRNA with two labels shows that RF classifiers are able to capture variations (cf. Figs. 2 and 3) that the extremely similar distributions of those two labels in Fig. 4 fail to display. This observation suggests that the global inhibition that is shown by the drop in the averages of both RNAup and IntaRNA scores is also a levelling phenomenon rather than a “linear” inhibition.

3.5 Additional Experiments

Impact of Boxes on Predictions. Among the five labels we are considering, two correspond to ncRNA classes defined by the presence of “boxes” in the sequence: C/D box snoRNAs and H/ACA box snoRNA. Those boxes are small and their consensus sequences are flexible (C: RUGAUGA, D: CUGA, H: ANANNA, ACA: ACA). Yet they might bias our results, especially those obtained from Kmer composition. To investigate this matter, we performed a brute force feature selection algorithm specific to random forest classifiers: the Boruta algorithm. This algorithm tests each feature, estimates its contribution to the classification and produces the list of features considered to be crucial for a given threshold of confidence (i.e. p-value, default = 0.05). Results show that only 50% or less of the critical features are compatible with a consensus sequence, even in the 2-label dataset (restrained to C/D box and H/ACA boxes snoRNAs only). This result is an upper bound since boxes are located while Kmer distributions ignore positions. As a consequence, the results displayed in Figs. 2 and 3 cannot be produced only from Kmers capturing boxes.

Impact of Sequence Lengths on RNA-RNA Interaction Predictions.

The third panel of Fig. 4 displays the distributions of lengths of ncRNAs for all labels, each label being represented by a boxplot in its usual colour. Length distributions vary from a label to another with two visible groups of labels: *C/D box*, *H/ACA box* and *spliceosomal* labels (resp. red, blue and yellow) distributions are tightened around a relatively short length while *miscellaneous* and *unknown* labels (resp. black and green) present a wider distribution with overall longer sequences. The problem planted by lengths of mRNAs targets has been explored by *Umu et al.* [19, 20]. Their results show that the accuracy of prediction tools typically drops as the length of the target increases above 300 nb. However, amongst the prediction tools tested, *IntaRNA* displays very little to no loss as the length of the target increases. On the contrary *RNAup* performances are significantly reduced. Cutting down the targets into subsequences of manageable length is not suited for this study as we need one score per {ncRNA,mRNA} pair. Moreover, we would like to propose to interpret this drop not only as a flaw of *RNAup* but as an illustration of the difference we described in Sect. 3.2. Yet the predictions scores for *miscellaneous* and *unknown* labels (resp. black and green) are to be treated with caution.

A second problem to consider is that the features we used are not independent of sequence lengths. Indeed, a longer sequence will contain more Kmers and Fig. 4 suggests a partial correlation between scores and length. In order to investigate this issue we repeated the ensemble learning procedure with the length as the only feature. Results show that predictions using length are accurate ($\mu = 0.856$ and $\sigma = 0.012$ with the 2-label dataset, $\mu = 0.651$ and $\sigma = 0.013$ with the 5-label dataset) but are slightly outperformed by the ones trained on *RNAup* scores over the 5-label dataset and over both datasets by the ones trained on Kmer composition. Those results suggest that sequence lengths are specific to each labels but are not the only variation captured by the classifiers.

Overlapping of Predicted Interaction Zones with Observed Interaction

Zones. We scanned the sequences of C/D box snoRNAs in the dataset looking for the consensus sequences. We excluded C/D box snoRNAs with ambiguous sites (i.e. more than one match with the consensus sequences of either box in the corresponding potential areas). We then looked for any intersection between the area interacting with rRNAs in observations (i.e. 3-rd to 11-th nucleotides upstream from D box) and the interaction zones predicted by *RNAup*. Amongst the interaction zones involving the 35 selected C/D box snoRNAs candidates, none overlapped with the observed interaction zones.

4 Conclusion

Our results enabled us to identify the signature of an evolutionary pressure against random interactions between ncRNAs and mRNAs in *Saccharomyces Cerevisiae*. Presumably, as previously observed in prokaryotes and archaea, this phenomenon aims to increase the translation efficiency [10].

Although our data set includes various types of ncRNAs, the vast majority of them are snoRNAs. Our conclusions are therefore primarily applicable to snoRNAs, even if our data do not exclude that it could be generalized to other ncRNAs. Interestingly, the (old) age of the snoRNA family suggests that it could be the trace of a fundamental biological process used by primitive microorganisms. The absence (to our knowledge) of experimental evidences of snoRNA-mRNA interactions in unicellular eukaryotes tends to support our conclusions. By contrast, the existence of known interactions between orphan snoRNAs and mRNAs in human or mice [4, 5] opens a legitimate debate about the necessity and specificity of such mechanisms in animals.

References

1. He, L., Hannon, G.J.: MicroRNAs: small RNAs with a big role in gene regulation. *Nat. Rev. Genet.* **5**(7), 522–531 (2004). <https://doi.org/10.1038/nrg1379>
2. Altuvia, S., Zhang, A., Argaman, L., Tiwari, A., Storz, G.: The *Escherichia coli* OxyS regulatory RNA represses *fhlA* translation by blocking ribosome binding. *EMBO J.* **17**(20), 6069–6075 (1998). <https://doi.org/10.1093/emboj/17.20.6069>
3. Scott, M.S., Ono, M.: From snoRNA to miRNA: dual function regulatory non-coding RNAs. *Biochimie* **93**(11), 1987–1992 (2011). <https://doi.org/10.1016/j.biochi.2011.05.026>
4. Sharma, E., Sterne-Weiler, T., O’Hanlon, D., Blencowe, B.J.: Global mapping of human RNA-RNA interactions. *Mol. Cell* **62**(4), 618–626 (2016). <https://doi.org/10.1016/j.molcel.2016.04.030>
5. Nguyen, T.C., Cao, X., Yu, P., Xiao, S., Lu, J., Biase, F.H., Sridhar, B., Huang, N., Zhang, K., Zhong, S.: Mapping RNA-RNA interactome and RNA structure in vivo by MARIO. *Nat. Commun.* **7**, 12023 (2016). <https://doi.org/10.1038/ncomms12023>
6. Panni, S., Prakash, A., Bateman, A., Orchard, S.: The yeast noncoding RNA interaction network. *RNA* **23**(10), 1479–1492 (2017). <https://doi.org/10.1261/rna.060996.117>
7. Aw, J.G.A., Shen, Y., Wilm, A., Sun, M., Lim, X.N., Boon, K.L., Tapsin, S., Chan, Y.S., Tan, C.P., Sim, A.Y.L., Zhang, T., Susanto, T.T., Fu, Z., Nagarajan, N., Wan, Y.: In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation. *Mol. Cell* **62**(4), 603–617 (2016). <https://doi.org/10.1016/j.molcel.2016.04.028>
8. Mattick, J.S.: RNA regulation: a new genetics? *Nat. Rev. Genet.* **5**(4), 316–323 (2004). <https://doi.org/10.1038/nrg1321>
9. Weill, N., Lisi, V., Scott, N., Dallaire, P., Pelloux, J., Major, F.: MiRBooking simulates the stoichiometric mode of action of microRNAs. *Nucleic Acids Res.* **43**(14), 6730–6738 (2015). <https://doi.org/10.1093/nar/gkv619>
10. Umu, S.U., Poole, A.M., Dobson, R.C., Gardner, P.P.: Avoidance of stochastic RNA interactions can be harnessed to control protein expression levels in bacteria and archaea. *Elife* **5** (2016). <https://doi.org/10.7554/eLife.13479>
11. Waters, L.S., Storz, G.: Regulatory RNAs in bacteria. *Cell* **136**(4), 615–628 (2009). <https://doi.org/10.1016/j.cell.2009.01.043>. <http://www.sciencedirect.com/science/article/pii/S0092867409001251>

12. Storz, G., Vogel, J., Wassarman, K.M.: Regulation by small RNAs in bacteria: expanding frontiers. *Mol. Cell* **43**(6), 880–891 (2011). <https://doi.org/10.1016/j.molcel.2011.08.022>. <http://www.sciencedirect.com/science/article/pii/S1097276511006435>
13. Sherman, D., Durrens, P., Beyne, E., Nikolski, M., Souciet, J.L.: Génolevures: comparative genomics and molecular evolution of hemiascomycetous yeasts. *Nucleic Acids Res.* **32**(Database Issue), D315–D318 (2004). <https://doi.org/10.1093/nar/gkh091>. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC308825/>
14. 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)
15. Wright, P.R., Georg, J., Mann, M., Sorescu, D.A., Richter, A.S., Lott, S., Kleinkauf, R., Hess, W.R., Backofen, R.: CopraRNA and IntaRNA: predicting small RNA targets, networks and interaction domains. *NAR* **42**(Web Server Issue), W119–W123 (2014). <https://doi.org/10.1093/nar/gku359>. PRW, JG and MM contributed equally to this
16. Thuriaux, P., Martin, C., Blondel, L., Visset, D.: Les organismes modèles: la levure. Belin, Paris (2004)
17. 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). <https://doi.org/10.1093/bioinformatics/btn544>
18. Tafer, H., Kehr, S., Hertel, J., Hofacker, I.L., Stadler, P.F.: RNAsnoop: efficient target prediction for H/ACA snoRNAs. *Bioinformatics* **26**(5), 610–616 (2010). <https://doi.org/10.1093/bioinformatics/btp680>
19. Lai, D., Meyer, I.M.: A comprehensive comparison of general RNA-RNA interaction prediction methods. *Nucleic Acids Res.* **44**(7), e61 (2016)
20. Umu, S.U., Gardner, P.P.: A comprehensive benchmark of RNA-RNA interaction prediction tools for all domains of life. *Bioinformatics* **33**(7), 988–996 (2017)
21. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*, 2nd edn. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**(Oct), 2825–2830 (2011)



Context-Specific Nested Effects Models

Yuriy Sverchkov¹(✉)() , Yi-Hsuan Ho²() , Audrey Gasch²() ,
and Mark Craven¹()

¹ Department of Biostatistics and Medical Informatics,
University of Wisconsin–Madison, Madison, WI, USA
yuriy.sverchkov@wisc.edu

² Department of Genetics, University of Wisconsin–Madison, Madison, WI, USA

Abstract. Advances in systems biology have made clear the importance of network models for capturing knowledge about complex relationships in gene regulation, metabolism, and cellular signaling. A common approach to uncovering biological networks involves performing perturbations on elements of the network, such as gene knockdown experiments, and measuring how the perturbation affects some reporter of the process under study. In this paper, we develop context-specific nested effects models (CSNEMs), an approach to inferring such networks that generalizes nested effect models (NEMs). The main contribution of this work is that CSNEMs explicitly model the participation of a gene in multiple *contexts*, meaning that a gene can appear in multiple places in the network. Biologically, the representation of regulators in multiple contexts may indicate that these regulators have distinct roles in different cellular compartments or cell cycle phases. We present an evaluation of the method on simulated data as well as on data from a study of the sodium chloride stress response in *Saccharomyces cerevisiae*.

1 Introduction

Cellular processes such as gene regulation, metabolism, and signaling form complex interplay of molecular interactions. A primary means of uncovering the details of these processes is through the analysis of measured responses of cells to perturbation experiments. We present Context-Specific Nested Effect Models (CSNEMs), which are graphical models for analyzing screens of high-dimensional phenotypes from gene perturbations. In this setting, the perturbation consists of knocking out, knocking down, or otherwise disabling the activity of a gene, via the use of deletion mutants, RNA interference, CRISPR/Cas9, or other techniques. The high-dimensional phenotype may be a transcriptomic, proteomic, metabolomic or similar multidimensional profile of measurements. Such profiles provide indirect information about the pathways that connect the gene that is perturbed in an experiment to the effects observed in a phenotype. This poses a challenge for determining functional relationships, since the precise mechanisms by which the perturbation relates to the phenotype must be inferred using computational and statistical methods, expert knowledge, or a combination of both.

Related work on inferring networks from gene expression data includes methods based on statistical dependencies between expression measurements [4, 7], which are used to construct networks of probable interactions between the genes measured in the expression profile. Other work on using phenotypic data uses clustering of phenotypic profiles, or the similarity between profiles, to construct networks among the perturbation genes [17, 19]. The rationale behind these approaches is that genes that produce similar phenotypes when perturbed are likely to be functionally related [13].

The CSNEM approach is a generalization of the Nested Effect Model (NEM) [11]. In the NEM approach, a network structure among the perturbed elements of the cell is inferred from the nested structure of phenotypic profiles. The general idea is that perturbation of a gene that is further upstream in a signaling pathway would affect more elements than perturbation of a gene further downstream. For example, Fig. 1(a) shows an NEM in which Hog1 is upstream of Cka2. The table underneath the graph represents the differential expressions of the high-dimensional phenotypes observed in the screen, with rows corresponding to single-gene knockouts and each column corresponding to an *effect*: one dimension of a phenotype, such as a particular transcript in a transcriptomic phenotype. In the table of effect measurements in the figure, a ‘1’ indicates that a perturbation changed the response of the effect, and a ‘0’ indicates that it did not. The deletion of Hog1 would affect e_1, e_2, e_3 and e_4 because they are all downstream of it. The deletion of Cka2, on the other hand, would only affect e_3 and e_4 . Therefore, the nesting of the effects of the deletion of Cka2 within the effects of the deletion of Hog1 places the former downstream of the latter.

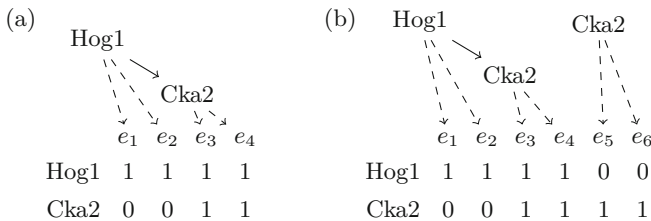


Fig. 1. (a) An example of effect nesting in an NEM, and (b) a partial intersection of effects as captured by a CSNEM. The table underneath each graph represents the differential expressions of the high-dimensional phenotypes observed in the screen, with rows corresponding to single-gene knockouts and each column corresponding to an *effect*, one dimension of a phenotype, where a ‘1’ indicates that a perturbation changed the response of the effect, and a ‘0’ indicates that it did not.

Such nesting of effects, however, does not always occur. The protein product of a gene may interact with those of other genes in a multitude of ways, and one might imagine a situation where two genes are interacting with each other upstream of a subset of the effects, but additionally have other roles independently of each other. This is the case in Fig. 1(b), where, upstream of effects e_1, e_2, e_3 and e_4 Cka2 and Hog1 interact as before, but Cka2 additionally affects

e_5 and e_6 independently of Hog1. In such a case, we see that the phenotype induced by the perturbations of each gene includes effects downstream of the common pathway, but each perturbation also shows unique effects, and rather than being nested, the effects show a partial intersection. The example in Fig. 1 is based on a pattern we identified in our application of CSNEM learning to experiments studying sodium chloride (NaCl) stress response in *Saccharomyces cerevisiae*.

In the CSNEM approach, we address this issue by explicitly considering the possibility that one gene may have multiple contexts of interaction. The model can be equivalently viewed either as a single graph model where multiple nodes may represent multiple roles of the same gene, or as a mixture of multiple NEMs, where each NEM describes a different subset of the effects. Notably, mixtures of NEMs have been used for analyzing single-cell expression data [22]. In that work, the mixture is used to account for variation of gene activation states across different cells. In contrast, in a CSNEM, the mixture represents different patterns of interaction among the same sets of genes across different subsets of the measured effects. The effect pattern in Fig. 1(b) can alternatively be accounted for by the introduction of a hidden node downstream of both Hog1 and Cka2, an approach explored by Sadeh et al. [21], where they introduce a statistical test to infer a partially resolved nested effect model. In fact, Sadeh et al. show that the presence of a hidden node downstream of a pair of genes is consistent with every possible configuration of effect responses. Their method aims to characterize all possible NEM models that are consistent with the data, and as a result it never rejects the possibility of a hidden node existing downstream of any pair of genes. In contrast, in our approach we aim to find a single parsimonious network model that optimally fits the data. We show how to cast the problem of learning a CSNEM as a modified version of NEM learning, evaluate the ability of this approach to recover a ground-truth network on simulated data, and present an application to the salt stress pathway in yeast.

2 Background: Nested Effects Models

Tresch and Markowitz [25] formulate nested effects models (NEMs) as a special case of effects models. In an effects model, there is a set of actions \mathcal{A} , and a set of effects \mathcal{E} , and we wish to model which effects change in response to each action. In earlier work on nested effects models [11], the actions and effects are respectively referred to as S-genes (S for signaling) and E-genes (E for effects). The actions correspond to perturbation experiments, while the effects correspond to the high-dimensional phenotype measured in the experiment. A general effects model can be represented by a binary matrix F where $F_{ae} = 1$ if action a leads to a response (or change) in effect e , and 0 otherwise.

Let $n_{\mathcal{A}}$ and $n_{\mathcal{E}}$ represent the number of actions and effects, respectively. An NEM is made up of a directed graph G the nodes of which are the actions \mathcal{A} , and an $n_{\mathcal{A}} \times n_{\mathcal{E}}$ binary matrix Θ of attachments, in which $\Theta_{ae} = 1$ if effect e is attached to action a , and 0 otherwise. A modeling constraint is that each effect is attached to at most one action.

The NEM is interpreted as follows: action a causes a response in effect e if and only if either e is attached directly to a , or there is a directed path in G from a to the action to which e is attached. Mathematically, this can be formulated in terms of matrix multiplication. Since what matters is which actions are reachable from other actions in G , we can work with Γ , the $n_A \times n_A$ accessibility matrix of G . Γ_{ab} is 1 if there is a directed path from a to b in G , and 0 otherwise. As a matter of convention and for mathematical convenience, the diagonal entries, Γ_{aa} are all 1s. Using Γ , we can express the effects matrix F of an NEM as $F = \Gamma\Theta$.

2.1 Likelihood Computation

The problem of inferring an NEM from a data set D can be viewed as that of maximizing a likelihood. In this section we review how the likelihood of an NEM is framed to illustrate how the likelihood of a CSNEM relates to it.

Supposing that we have some data consisting of measurements of the observable effects subject to each action included in the model, and assuming data independence, for a general effects model, the log-likelihood of the model is

$$\log L(F) = \log \mathbb{P}(D|F) = \sum_{(a,e) \in \mathcal{A} \times \mathcal{E}} \log \mathbb{P}(D_{ae}|F_{ae}). \quad (1)$$

Where $\mathbb{P}(D_{ae}|F_{ae})$ is the probability of the data we observed in regard to effect e subject to action a given that F_{ae} indicates whether we expect a response in e subject to a . When the observed phenotype is, for example, gene expression data, a typical indicator of a response in effect e is differential expression of effect e between the experimental condition a and a control, such as a wild-type phenotype.

Let $R \in \mathbb{R}^{n_{\mathcal{E}} \times n_{\mathcal{A}}}$ be a matrix of log-likelihood ratios such that $R_{ea} = \frac{\mathbb{P}(D_{ae}|F_{ae}=1)}{\mathbb{P}(D_{ae}|F_{ae}=0)}$, and let N represent the null model predicting no effect response to any action, Tresch and Markowitz [25] show that the log-likelihood of an effects model F is then

$$\log L(F) = \text{tr}(FR) + \underbrace{\log L(N)}_{\text{constant w.r.t. data}} \quad (2)$$

where $\text{tr}(\cdot)$ is the trace of a matrix. The above holds for any effects model in general. Since in an NEM, $F = \Gamma\Theta$, to maximize the likelihood of an NEM one would maximize $\text{tr}(\Gamma\Theta R)$.

Computationally, maximizing this expression is difficult because it is a search over a discrete but exponentially large space of all possible Γ and Θ matrices. Early work on NEMs reduces some of the complexity of this search by observing that since Θ can only have one 1 for each effect across all actions by construction, and since $\text{tr}(\Gamma\Theta R) = \text{tr}(R\Gamma\Theta)$, one can marginalize over all possible values of Θ , assuming that they are equally likely *a-priori*, yielding a marginal likelihood proportional to $\prod_{e \in \mathcal{E}} \sum_{a \in \mathcal{A}} \exp((R\Gamma)_{ea})$. This reduces the task to the search for a Γ that maximizes this marginal likelihood, an exhaustive search for which

is feasible for $n_{\mathcal{A}} \leq 5$ [11]. For larger graphs, however, the problem is still computationally restrictive, and multiple algorithms for learning nested effects model structure efficiently have been presented in the literature [6, 12], most of which have been implemented in the `nem` R package [5]. Other approaches to computing the likelihood have also been explored, such as the factor graph optimization approach by [26].

In this work, we show how learning a CSNEM can be cast as a more complex NEM learning problem. To solve the NEM learning problem, we use MC-EMiNEM, a method that does not attempt to optimize a marginal likelihood, as many of the above approaches do, but maximizes the log posterior

$$\log \mathbb{P}(I, \Theta | D) = \log L(I\Theta) + \sum_{(a,b) \in \mathcal{A} \times \mathcal{A}} \log \mathbb{P}(\Gamma_{a,b}) + \log \mathbb{P}(\Theta). \quad (3)$$

Where $\log \mathbb{P}(\Gamma_{i,i})$ is an edge-wise prior on the structure of the actions graph and $\mathbb{P}(\Theta)$ is a prior on the attachment matrix. MC-EMiNEM uses Monte Carlo (MC) sampling and Expectation Maximization (EM) within MC steps to search for the I and Θ that are optimal with respect to this posterior [16]. MC-EMiNEM is available as a part of the `nem` R package.

3 Methods: Context-Specific Nested Effects Models

As briefly mentioned in the introduction, the motivation for developing CSNEMs is that there are cases in which phenotype effects are not nested, as in the example in Fig. 1. In CSNEMs, we account for situations like the partial overlap in Fig. 1 by allowing an action in the graph to be represented by more than one node, and we call these different nodes that correspond to the same action different *contexts* of the action. Mathematically, this enables the model to represent relationships that are not representable by an NEM. Biologically, different contexts in a CSNEM may correspond to participation in different pathways, either due to physical separation such as localization of molecules, or temporal separation, such as participation in different stages of the cell cycle.

The CSNEM in Fig. 1(b) is presented as a single NEM-like graph with multiple contexts for the Cka2 node. Note that the same diagram can also be viewed as a pair of NEMs: one containing Hog1 and Cka2, which applies to effects e_1, e_2, e_3, e_4 , and another containing only Cka2, which applies to the effects e_5 and e_6 . This view of a CSNEM as a mixture of NEMs is most useful in understanding our approach to learning a CSNEM from data.

3.1 The Likelihood of a k -CSNEM

We define a k -CSNEM as a mixture of k NEM's, where the response of each effect e is governed by one of k NEMs, each of which can have a different graph G relating the actions \mathcal{A} . A k -CSNEM is therefore parameterized by k accessibility matrices $\Gamma^1, \dots, \Gamma^k$, each of which is $n_{\mathcal{A}} \times n_{\mathcal{A}}$ and by a vector θ , each coordinate

of which takes one of $kn_{\mathcal{A}} + 1$ values, specifying attachment to one of the $n_{\mathcal{A}}$ actions in one of the k NEMs, or the absence of attachment.

The parameter θ partitions the space of effects by assigning each effect to one of the k NEMs (or to none of them). As a matter of convention, we represent attachment of effect $e \in \mathcal{E}$ to an action $a \in \mathcal{A}$ in mixture member $i \in \{1, \dots, k\}$ by $\theta_e = (i - 1)n_{\mathcal{A}} + a$ (we slightly abuse notation, treating actions as natural numbers $1, \dots, n_{\mathcal{A}}$ here), and let $\theta_e = 0$ if the effect is not attached to any action in any NEM. We can then define the partition of \mathcal{E} into k sets $\mathcal{E}_1, \dots, \mathcal{E}_k$ as

$$\mathcal{E}_i = \{e \in \mathcal{E} \mid \exists a \in \mathcal{A} : \theta_e = (i - 1)|\mathcal{A}| + a\} \text{ for } i \in \{1, \dots, k\}. \quad (4)$$

Let us define a mapping of effect indices, which will be useful later: $\zeta : \{1, \dots, k\} \times \{1, \dots, |\mathcal{E}_i|\} \rightarrow \mathcal{E}$. Thus, $\zeta(i, j) = e$ when effect e is the j th member of partition \mathcal{E}_i . Given this partition, the likelihood of a CSNEM is defined as the product of the NEM likelihoods per partition:

$$L(\Gamma^{1, \dots, k}, \theta) = \prod_{i=1}^k L(\Gamma^i, \Theta^i) \quad (5)$$

where Θ^i is a matrix in $\{0, 1\}^{|\mathcal{A}| \times |\mathcal{E}_i|}$ and $\Theta_{aj}^i = 1$ iff $\theta_{\zeta(i, j)} = (i - 1) + a$, and 0 otherwise.

In relation to the CSNEM, let us combine the mixture of NEMs into one structure by defining the block diagonal matrix Γ made of blocks Γ^i , define $\Theta \in \{0, 1\}^{|\mathcal{A}| \times |\mathcal{E}|}$ by $\Theta_{ae} = 1$ iff $\theta_e = a$, and let be a block matrix made up of k appended $|\mathcal{A}| \times |\mathcal{A}|$ identity matrices:

$$\Gamma = \begin{bmatrix} \Gamma^1 & 0 & \dots & 0 \\ 0 & \Gamma^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \Gamma^k \end{bmatrix}, \quad \Psi = \underbrace{[I_{|\mathcal{A}|} \ I_{|\mathcal{A}|} \ \dots \ I_{|\mathcal{A}|}]}_{k \text{ copies}}. \quad (6)$$

Let R^i be a matrix in $\mathbb{R}^{|\mathcal{E}_i| \times |\mathcal{A}|}$ where $R_{ja}^i = R_{\zeta(i, j), a}$ (i.e., R^i is a selection of effects from R based on the partition \mathcal{E}_i). Given these definitions the log-likelihood of the CSNEM can be written as¹

$$\log \prod_{i=1}^k L(\Gamma^i, \Theta^i) = \text{tr}(\Gamma \Theta (R \Psi)) + \log L(N). \quad (7)$$

Thus, the likelihood of a k -CSNEM is equal to the likelihood of an NEM with $k|\mathcal{A}|$ actions for the data matrix $R\Psi$, subject to the constraint that Γ is block diagonal as in (6). We can consequently use any NEM learner to learn a k -CSNEM mixture, as long as it supports constraining Γ to be block-diagonal.

¹ For a detailed derivation see <https://github.com/sverchkov/mc-em-cs-nem/blob/master/recomb-2018-supplement/recomb-2018-supplement.pdf>, commit 98b01f19357e3d58eae81764d42a6903624e3433 at the time of submission.

Analogously to (3), we can obtain a posterior probability for the CSNEM by introducing priors for Γ and Θ , and applying MC-EMiNEM to maximize that posterior. The block-diagonal constraint can be enforced using the edge-wise prior on the structure of Γ , by setting the priors on edges that would violate block-diagonality to zero.

3.2 Compact Visualization and Identifiability of a k -CSNEM

Having obtained k NEMs and the corresponding partitioning of the effect set, a single graph can be composed by merging all action nodes across the graphs that have the same ancestors (are reachable from the same set of actions). Figure 2 provides an example: Fig. 2(a) shows three graphs that describe the structures of three NEMs that compose a mixture, and Fig. 2(b) shows the result of merging them. Note that Hog1 is reachable from no nodes but itself in all three NEMs. Consequently, in the compact CSNEM, there is only one version of Hog1. In contrast, Cka2 is reachable from Hog1 in one of the NEMs, and is only reachable from itself in the others, which is why it has two contexts in the CSNEM. Similarly, Ckb14 is reachable from both Hog1 and Cka2 in one of the three NEMs, but not the others, and has two contexts as well. To keep track of the various contexts, we append the list of genes from which a context is reachable when displaying the graph, e.g. the context of Cka2 that is reachable from Hog1 is labeled ‘Cka2 [Hog1],’ while the context that is not reachable from other nodes is labeled simply ‘Cka2.’ This is particularly helpful when viewing graphs with many nodes and many contexts.

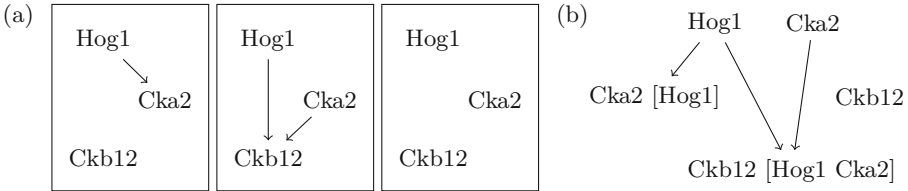


Fig. 2. Building a CSNEM from a mixture of NEMs. (a) Three NEMs that compose a mixture. (b) A single graph obtained by an edge-preserving merge of the three NEMs.

The merged graph in Fig. 2(b) preserves the edges that were present in the mixture of NEMs, but it is not necessarily a unique maximizer of the likelihood, rather, it is a member of an equivalence class of equally likely CSNEMs. What characterizes the equivalence class is the set of *inclusive ancestries* of the nodes in the CSNEM. The inclusive ancestry of a node is a set of actions; this set contains the action at the node and all actions from which it is reachable: e.g. the inclusive ancestry of the Cka2 node in the leftmost NEM in Fig. 2(a) is {Hog1, Cka2}, while the inclusive ancestry of the Cka2 node in the middle NEM is simply {Cka2}. The set of inclusive ancestries for the example in Fig. 2 is

therefore $\{\{\text{Hog1}\}, \{\text{Cka2}\}, \{\text{Ckb12}\}, \{\text{Hog1}, \text{Cka2}\}, \{\text{Hog1}, \text{Cka2}, \text{Ckb12}\}\}$. Any two CSNEMs with identical sets of inclusive ancestries necessarily have the same set of unique accessibility matrix columns $\Gamma_a^i : i \in \{1, \dots, k\}, a \in \mathcal{A}$, and consequently, have the same likelihood for likelihood-maximizing attachments Θ . The characterization of equivalence classes in terms of inclusive ancestry sets relates to previous results about NEM identifiability: for transitively closed Γ , cycles form fully connected components that can be merged into single nodes [12]. All nodes in such connected components have identical ancestry sets, yielding a one-to-one mapping from the NEM’s nodes to the ancestry sets, where the edges in the transitive closure of the NEM correspond to the set inclusion relations between ancestry sets. This can also be extended to the case of non-transitive Γ and the result on identifiability of non-transitive NEMs up to cycle reversals [25], the full discussion of which we omit here for brevity. Note that while set of ancestries characterizes the likelihood equivalence class, the posterior maximized by MC-EMiNEM would be, for example, higher for CSNEMs with fewer edges in Γ under a sparsifying edge prior.

4 Results

We have introduced the CSNEM model and showed how the CSNEM likelihood can be viewed as the likelihood of an NEM with $kn_{\mathcal{A}}$ actions learned from a modified differential expression log-likelihood ratio matrix $R\Phi$. Below, we use this transformation in conjunction with an existing NEM learning approach, MC-EMiNEM to learn CSNEMs and evaluate the ability of this approach to recover a CSNEM from data that is generated by a known multiple-context model in simulation. Finally, we present the results of learning a CSNEM from the results of knockout experiments on *S. cerevisiae* cells under NaCl stress, and discuss the biological significance of some patterns of context-specificity that are identified in the CSNEM.

4.1 Evaluation on Simulated Data

We performed simulations to evaluate our ability to infer CSNEMs from data. We generated data from mixtures of NEMs of varying size: we varied the size of the NEMs in the mixture to contain $n_{\mathcal{A}} = 3, 5, 10$, or 20 actions, and we varied the number of NEMs in the generating model from $j = 1$ to $j = 5$, inclusive, with $j = 1$ being equivalent to a simple NEM model. The number of effects $n_{\mathcal{E}}$ was fixed at 1000. We generated 30 mixtures corresponding to each configuration of j and $n_{\mathcal{A}}$, resulting in a total of 600 generated models. To generate each mixture, first we generated j random directed graphs G_1, \dots, G_j of $n_{\mathcal{A}}$ nodes, by drawing each of the possible $n_{\mathcal{A}}^2$ edges of the graph with a probability of 0.2 for graphs of size $n_{\mathcal{A}} < 20$ and a probability of 0.04 for graphs of size $n_{\mathcal{A}} = 20$ (with the higher edge density of 0.2 for 20 nodes, all nodes become reachable from all other nodes, yielding degenerate effect patterns where each effect is either affected by all actions, or by none). Next, for each effect, with probability 0.3 we attach it

nowhere, otherwise, we uniformly randomly attach it to one of the $n_{\mathcal{A}} \times j$ nodes in all of these graphs. Given these graphs and effect attachments, we infer which effects are reachable from each node, and compute the $n_{\mathcal{A}} \times n_{\mathcal{E}}$ binary effect matrix F^T , where $F_{as}^T = 1$ if and only if effect s is reachable from action a in any one of the j graphs. Next, we generate a log-odds matrix that represents a noisy measurement of this effect matrix by drawing from $\log \frac{\text{Beta}(\beta, 1)}{\text{Beta}(1, \beta)}$ for each ‘true’ cell and from $\log \frac{\text{Beta}(1, \beta)}{\text{Beta}(\beta, 1)}$ for each ‘false’ cell, with $\beta = 10$. This process generates the log-odds matrix R that we use as input to our learning method. Additionally, to examine the effect of noise in the measurement of effects on model inference, we generated log-odds matrices using $\beta = 1, 2, 5$ from the first 10 generating mixtures with $n_{\mathcal{A}} = 20, j = 1, 3, 5$.

Since in real-world applications we usually do not know how many contexts are truly needed to describe a process under study, we sweep through values of k ranging from 1 to 8, and learn a k -CSNEM for each value of k from each generated log-odds matrix. CSNEMs were learned using the MC-EMiNEM implementation in the `nem` R package, with the learned network taken from the end of a 20000 sample chain, the empirical Bayes step performed every 5000 steps, an acceptance sparsity prior of 0.5, and $kn_{\mathcal{A}}$ edges changed in every MCMC step (see Niederberger et al. [16] for details on how these settings are used in MC-EMiNEM). The edge-wise prior for permissible edges was set to 0.2.

We evaluate each k -CSNEM learned from each log-odds matrix both in terms of the ability of the CSNEM to accurately model which effects are differentially expressed in response to each action and in terms of the relationships inferred among actions. In the former case, we use the F-measure to quantify how well the effect matrix F of the learned CSNEM matches that of the generating CSNEM, with the interpretation that if an effect responds to an action in both the learned and the generating model, it is a true positive, if it doesn’t respond in the learned model but does in the generating model it is a false negative, if it doesn’t respond in either model it is a true negative, and if it responds in the learned model but not the generating model it is a false positive. Figure 3(a) shows the F-measures for learning the effect matrix across our simulations for the almost-noiseless case of $\beta = 10$. Figure 3(c) shows the F-measures for learning the effect matrix of a 20-action network from log-odds matrices generated with varying settings of β .

To compare the learned graph structures to the generating graph structures, we must first determine which contexts in the learned model correspond to which contexts in the generating model. For each action a in each model, we obtain a list of contexts that are distinguishable in terms of which actions are ancestors of the action a . We then match each of these contexts in each model to their best match in the other model. Each ancestor that the two contexts in the best match have in common counts as a true positive, each ancestor that appears in the context from the learned model but not in the context from the generating model counts as a false positive, and each ancestor that appears in the context from the generating model but not in the context from the true model counts as a false negative. We use these counts to summarize agreement between the

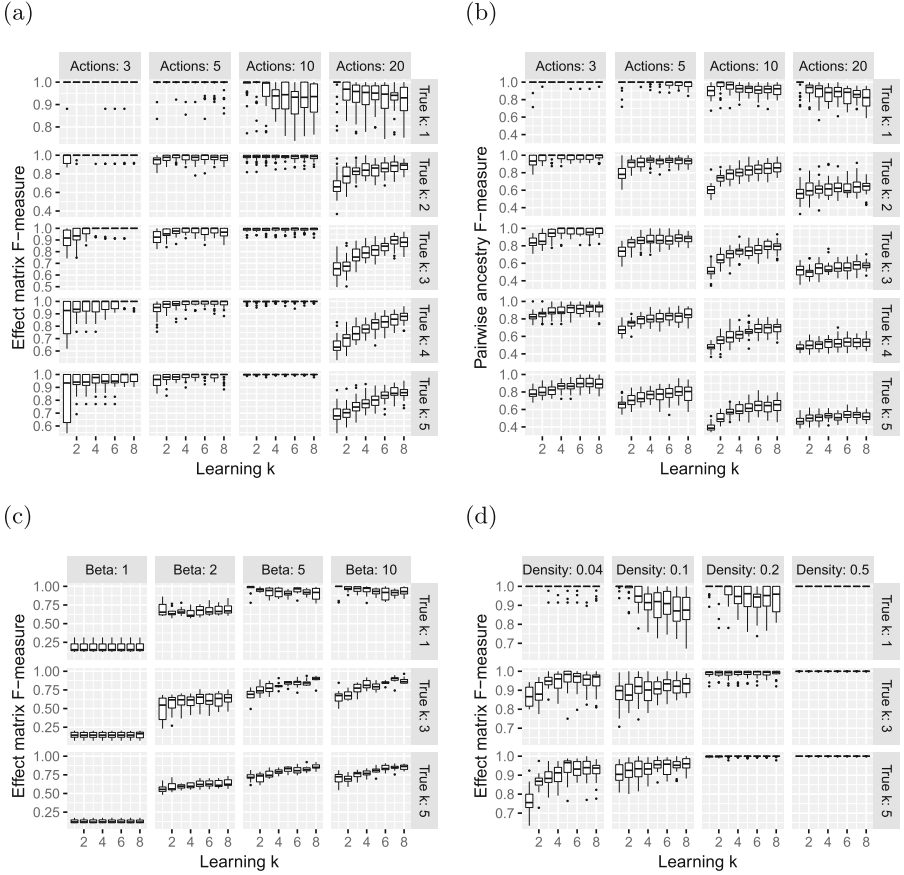


Fig. 3. Box plots of simulation F-measures. Each plot represents an aggregate of results from 30 random simulation replicates. Grid rows correspond to the number of contexts in the generating model, the x -axis in each of the grid cells indicates the number of contexts in the learned model, and the y -axis represents: (a) the F-measure of recovering the generating model’s effect matrix from the learned model across different sizes of action sets (grid columns) from log-odds matrices generated with $\beta = 10$, (b) the F-measure of recovering ancestry relationships, (c) the F-measure of learning the effect matrix of a 20-action network from log-odds matrices generated with varying settings of β (grid columns), and (d) the F-measure of learning the effect matrix from 10-action networks of varying density (grid columns) with log-odds generated using $\beta = 10$.

structures of two CSNEMs in terms of an F-measure which we call the pairwise ancestry F-measure. Figure 3(b) shows the pairwise ancestry F-measures across our simulations.

When the learned model is a plain NEM ($k = 1$), we see that as the generating model has more contexts, the recovery of both the effect and the ancestry pattern worsens (with the exception of the 10 actions case, examined below). This confirms that a CSNEM is necessary when multiple contexts are indeed in

play in the generating system. When the learned model has multiple contexts, even when the number of contexts in the learned model exceeds the number of contexts in the generating model, the approach does not seem to be susceptible to overfitting. This pattern holds as we increase noise (decrease β) in data generation.

At $n_{\mathcal{A}} = 10$ the NEM appears to recover the effects patterns well even when there are multiple contexts in the generating models, and we hypothesize that this is because of high connectivity in those ground truth networks: the average in-degree and out-degree of node is the product of one less than the number of actions times the edge density. We generated 20 mixtures for varying node densities (0.04, 0.1, 0.2, 0.5) with $j = 1, 3, 5$ contexts and $n_{\mathcal{A}} = 10$ nodes, and examined the effect-matrix F-measures across densities (Fig. 3(d)). Denser networks are perfectly recovered by single-context NEMs; this is likely because denser networks are more likely to lead to fully-connected transitive reductions, reducing the number of unique response patterns of effects, yielding data that is easier to capture in a simple NEM model. When the generating models are not too dense, CSNEMs are better than NEMs at recovering the effect patterns generated from multiple-contexts.

4.2 Application to NaCl Stress Response in *S. cerevisiae*

We apply our method to the exploration of NaCl stress response pathways in *S. cerevisiae*. We consider data obtained from a wild-type (WT) strain and 28 knockout strains. Transcript abundances were measured by microarray for each strain prior to NaCl treatment and 30 min after 0.7 M NaCl treatment. The data collection was described in detail in previous work [1, 8].

We are interested in how the gene knockouts change the cells' response to stress. Therefore, the actions \mathcal{A} in our model correspond to the knockouts. Since we use microarray data, the observations \mathcal{E} correspond to transcripts. The change in response is quantified as a change in log-fold-change. For each strain, we have the log-fold-change of transcript abundances in the sample 30 min after NaCl treatment as compared to the abundances in the sample prior to treatment. We then consider the difference between the log-fold-change in each knockout strain and that in the wild-type strain. To obtain the log-odds matrix R we use an empirical Bayes method to obtain log-posterior-odds of differential expression [10, 24] which is implemented in the `limma` R package [23]. Figure 4 shows the 3-CSNEM that was learned from the data.² The MC-EMiNEM settings used for learning both of these models are the same as those used for learning in the simulation experiments.

The inferred network captures many known and several new features of the yeast stress responsive signaling network. The Hog1 kinase is a master regulator of the osmotic stress response [15]. The CSNEM network correctly places Hog1 at the top of the hierarchy in paths with known co-regulators. For example, the

² An NEM learned from the data is at <https://github.com/sverchkov/mc-em-cs-nem/blob/master/recomb-2018-supplement/recomb-2018-supplement.pdf>.

network captures paths containing Hog1 and CK2 complex subunits Cka2 and Ckb1/2—Hog1 is known to interact physically with Cka2, and the two kinases regulate an overlapping set of genes [3]. The network also correctly predicts that the transcription factor Msn2 is regulated by Hog1, Pde2, and Snf1—all known regulators of Msn2 [9, 14, 18, 20]; yet a separate branch represents only Pde2 and Msn2, consistent with Pde2 playing a more significant role in regulating this transcription factor during salt stress [3]. Another example is seen in YGR122W, a poorly characterized protein required for processing the transcriptional repressor Rim101—the CSNEM correctly puts YGR122W and Rim101 in the same paths, with at least one regulatory branch shared with Hog1 control.

The CSNEM naturally produces groups of effects where each group comprises those effects (i.e. transcripts) that are reachable from contexts of actions in the graph. We examined the groups of effects in terms of Gene Ontology (GO) enrichments. Figure 5 shows a comparison of these enrichments to those obtained from grouping effects by the attachments from a learned NEM. The figure also shows a coarser split of the effects into groups based on CSNEM contexts: if an action was merged from two or more contexts in the single-network CSNEM representation, all the effects attached to it are considered reachable from both (or all three) contexts from which the action was merged. Each column in the figure corresponds to a GO term and each row corresponds to a combination of contexts or an action. A point in the figure indicates that the set of effects reachable from the context(s) or action was found to be significantly enriched for the GO term. Significance was defined according to a hypergeometric test with the Benjamini-Hochberg method used to control the false discovery rate at 0.05; only groups of five or more effects were considered for enrichment analysis.

A key advantage of our approach is that regulators can be represented in multiple pathways, capturing regulators that may have distinct roles in different cellular compartments or cell cycle phases. In fact, several of the GO terms for which the CSNEM effect groups are enriched are associated with subcellular localization and include transcripts encoding proteins localized to the nucleus, nucleolus, plasma membrane, endoplasmic reticulum, mitochondria, peroxisome, and cytoskeleton. The coarser split of effects by contexts also shows that there are clear divisions of localization across contexts in the CSNEM.

An interesting example of the benefits of the CSNEM approach is seen in its ability to capture the disparate signaling roles of the phosphatase Cdc14, a key regulator of mitotic progression in dividing cells [27]. Inactive Cdc14 is tethered to the nucleolus during much of the cell cycle but released upon mitosis to other subcellular regions where it dephosphorylates cyclins and other targets [28]. Separate from its role in the cell cycle, Cdc14 was recently linked to the stress response in yeast [2, 3], although its precise role is not clear.

The CSNEM network places Cdc14 in multiple pathways that capture the distinct functions of the phosphatase. One path represents an isolated connection of Cdc14 to a group of genes regulated by the cell cycle network. Many of these genes are known to be regulated by Cdc14 during normal cell cycle progression. But consistent with a second role in the stress response, Cdc14 is

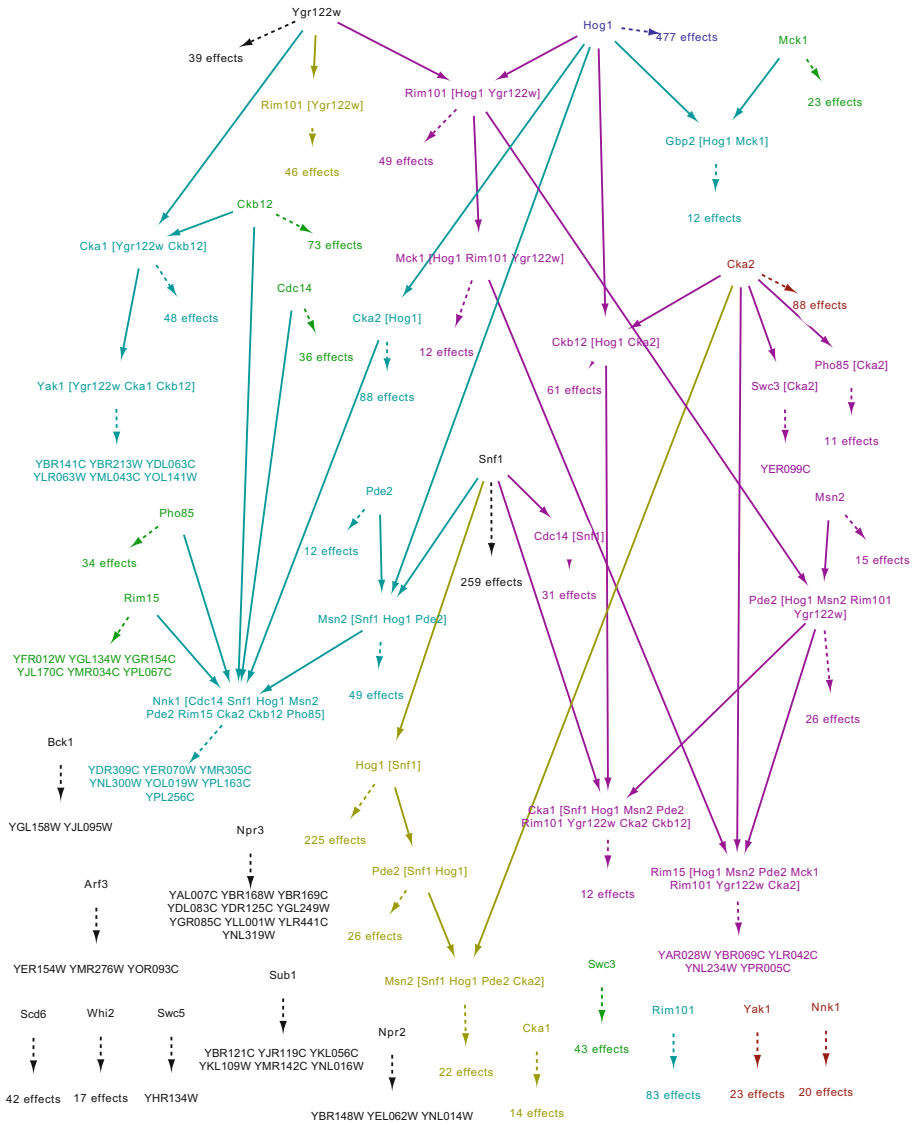


Fig. 4. The 3-CSNEM network learned from *S. cerevisiae* NaCl stress knockout microarray data. Action nodes and action-action edges are colored according to the NEM member in the mixture from which they came, in cyan, magenta, or yellow. Nodes that were merged because of identical ancestors in multiple mixture members are colored according to subtractive color mixing (cyan and magenta make blue, cyan and yellow make green, magenta and yellow make red, and all three make black). Effects are colored and grouped according to the actions to which they are attached. Where the number of effects in a group is fewer than 10, the effects are listed. Where it is 10 or more, the number of effects in the group is shown. Action-action edges are solid and action-effect edges are dashed. (Color figure online)

also nested in a path regulated by Snf1, a kinase that responds to both nutrient/energy restriction and osmotic stress resulting from salt treatment [29]. The Snf1-Cdc14 pathway is connected to 31 effectors that include genes induced by stress and related to glucose metabolism. Work from the Gasch Lab previously showed through genetic analysis that Snf1 and Cdc14 function, at least in part, in the same pathway during the response to salt stress [3]. Yet both Cdc14 and Snf1 have other functions in the cell, leading to the regulation of only partially overlapping gene sets. Thus, the CSNEM approach successfully captured this complex regulatory distinction for Cdc14 and Snf1.

5 Discussion

We have introduced CSNEMs, a generalization of NEMs which can explicitly model the different interactions that genes may have in different contexts. We have shown how a CSNEM can be viewed as a mixture of NEMs, and that the task of learning such a mixture can be cast as a single NEM-learning task with a modified data matrix and constrained action graph structure in which actions are replicated k times. Particularly, we took the approach of using a hard mixture where effects and actions are assigned to different contexts. A natural avenue for future investigation would be the exploration of soft-mixture approaches, which may prove more scalable for larger numbers of contexts and actions.

Applying our method to simulated data has shown that learning CSNEMs leads to good recovery of the effect patterns and ancestry relations that were present in the generating model. The results also show that a CSNEM is necessary when the generating model truly has multiple contexts, but slight over- or underestimation of the number of contexts does not seem to lead to overfitting. In practice, the correct number of contexts that a learned model should have is not known, and optimal selection of k is still an open problem that we plan to explore in future work. Existing approaches to model selection, such as a search for a plateau in likelihood or the use of model complexity measures such as AIC point to possible solutions to this problem.

Our analysis of a CSNEM network learned from *S. cerevisiae* NaCl-stress knockout microarray data revealed that the CSNEM does recover known regulatory patterns and moreover, captures known patterns of context-specificity in the genes under study. Analysis of GO term enrichments of the effects reachable from CSNEM nodes shows that many effect groups are associated with subcellular localization, a pattern even more evident in examining a coarser division of the effects, based on mixture contexts. We believe that localization may be one source of context-specificity that is relevant in many applications. The main motivation for developing CSNEMs was the observation that effect nesting may not be an appropriate assumption for some settings because of the context-specific nature of interactions that some genes can have, and perhaps more explicit modeling of contexts of interaction can lead to more faithful representations of the underlying biology.

Acknowledgments. We thank anonymous reviewers for many constructive comments. This research was supported by NIH/NLM grant T15 LM0007359, NIH/NIAID grant U54 AI117954, and NIH/NIGMS grant R01 GM083989.

References

1. Berry, D.B., Gasch, A.P.: Stress-activated genomic expression changes serve a preparative role for impending stress in yeast. *Mol. Biol. Cell* **19**(11), 4580–4587 (2008)
2. Breitskreutz, A., Choi, H., Sharom, J.R., Boucher, L., Neduva, V., Larsen, B., Lin, Z.-Y., Breitskreutz, B.-J., Stark, C., Liu, G.: A global protein kinase and phosphatase interaction network in yeast. *Science* **328**(5981), 1043–1046 (2010)
3. Chasman, D., Ho, Y.-H., Berry, D.B., Nemecek, C.M., MacGilvray, M.E., Hose, J., Merrill, A.E., Lee, M.V., Will, J.L., Coon, J.J.: Pathway connectivity and signaling coordination in the yeast stress-activated signaling network. *Mol. Syst. Biol.* **10**(11), 759 (2014)
4. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7**(3–4), 601–620 (2000)
5. Fröhlich, H., Beißbarth, T., Tresch, A., Kostka, D., Jacob, J., Spang, R., Markowetz, F.: Analyzing gene perturbation screens with nested effects models in R and bioconductor. *Bioinformatics* **24**(21), 2549–2550 (2008)
6. Fröhlich, H., Fellmann, M., Sültmann, H., Poustka, A., Beißbarth, T.: Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinform.* **8**(1), 1 (2007)
7. Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **5**(9), e12776 (2010)
8. Lee, M.V., Topper, S.E., Hubler, S.L., Hose, J., Wenger, C.D., Coon, J.J., Gasch, A.P.: A dynamic model of proteome changes reveals new roles for transcript alteration in yeast. *Mol. Syst. Biol.* **7**(1), 514 (2011)
9. Lee, P., Cho, B.-R., Joo, H.-S., Hahn, J.-S.: Yeast Yak1 kinase, a bridge between PKA and stress-responsive transcription factors, Hsf1 and Msn2/Msn4. *Mol. Microbiol.* **70**(4), 882–895 (2008)
10. Lönnstedt, I., Speed, T.: Replicated microarray data. *Statistica Sinica* **12**(1), 31–46 (2002)
11. Markowetz, F., Bloch, J., Spang, R.: Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics* **21**(21), 4026–4032 (2005)
12. Markowetz, F., Kostka, D., Troyanskaya, O.G., Spang, R.: Nested effects models for high-dimensional phenotyping screens. *Bioinformatics* **23**(13), i305–i312 (2007)
13. Markowetz, F., Spang, R.: Inferring cellular networks a review. *BMC Bioinform.* **8**(6), S5 (2007)
14. Mayordomo, I., Estruch, F., Sanz, P.: Convergence of the target of rapamycin and the Snf1 protein kinase pathways in the regulation of the subcellular localization of Msn2, a transcriptional activator of STRE (Stress Response Element)-regulated genes. *J. Biol. Chem.* **277**(38), 35650–35656 (2002)
15. Nadal, E., Posas, F.: Osmostress-induced gene expression—a model to understand how stress-activated protein kinases (SAPKs) regulate transcription. *FEBS J.* **282**(17), 3275–3285 (2015)

16. Niederberger, T., Etzold, S., Lidschreiber, M., Maier, K.C., Martin, D.E., Frohlich, H., Cramer, P., Tresch, A.: MC EMiNEM maps the interaction landscape of the mediator. *PLoS Comput. Biol.* **8**(6), e1002568 (2012)
17. Ohya, Y., Sese, J., Yukawa, M., Sano, F., Nakatani, Y., Saito, T.L., Saka, A., Fukuda, T., Ishihara, S., Oka, S.: High-dimensional and large-scale phenotyping of yeast mutants. *Proc. Nat. Acad. Sci. U.S.A.* **102**(52), 19015–19020 (2005)
18. Petrenko, N., Chereji, R.V., McClean, M.N., Morozov, A.V., Broach, J.R.: Noise and interlocking signaling pathways promote distinct transcription factor dynamics in response to different stresses. *Mol. Biol. Cell* **24**(12), 2045–2057 (2013)
19. Piano, F., Schetter, A.J., Morton, D.G., Gunsalus, K.C., Reinke, V., Kim, S.K., Kemphues, K.J.: Gene clustering based on RNAi phenotypes of ovary-enriched genes in *C. elegans*. *Curr. Biol.* **12**(22), 1959–1964 (2002)
20. Rep, M., Krantz, M., Thevelein, J.M., Hohmann, S.: The transcriptional response of *Saccharomyces cerevisiae* to osmotic shock Hot1p and Msn2p/Msn4p are required for the induction of subsets of high osmolarity glycerol pathway-dependent genes. *J. Biol. Chem.* **275**(12), 8290–8300 (2000)
21. Sadeh, M.J., Moa, G., Spang, R.: Considering unknown unknowns: reconstruction of nonconfoundable causal relations in biological networks. *J. Comput. Biol.* **20**(11), 920–932 (2013)
22. Siebourg-Polster, J., Mudrak, D., Emmenlauer, M., Rämö, P., Dehio, C., Greber, U., Fröhlich, H., Beerenwinkel, N.: NEMix: single-cell nested effects models for probabilistic pathway stimulation. *PLoS Comput. Biol.* **11**(4), e1004078 (2015)
23. Smyth, G.K.: Limma: linear models for microarray data. In: Gentleman, R., Carey, V.J., Huber, W., Irizarry, R.A., Dudoit, S. (eds.) *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, New York (2005). <https://doi.org/10.1007/0-387-29362-0-23>
24. Smyth, G.K.: Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat. Appl. Genet. Mol. Biol.* **3**(1), 3 (2004)
25. Tresch, A., Markowitz, F.: Structure learning in nested effects models. *Stat. Appl. Genet. Mol. Biol.* **7**(1), 9 (2008)
26. Vaske, C.J., House, C., Luu, T., Frank, B., Yeang, C.-H., Lee, N.H., Stuart, J.M.: A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Comput. Biol.* **5**(1), e1000274 (2009)
27. Weiss, E.L.: Mitotic exit and separation of mother and daughter cells. *Genetics* **192**(4), 1165–1202 (2012)
28. Wurzenberger, C., Gerlich, D.W.: Phosphatases: providing safe passage through mitotic exit. *Nat. Rev. Mol. Cell Biol.* **12**(8), 469–482 (2011)
29. Ye, T., Elbing, K., Hohmann, S.: The pathway by which the yeast protein kinase Snf1p controls acquisition of sodium tolerance is different from that mediating glucose regulation. *Microbiology* **154**(9), 2814–2826 (2008)



Algorithmic Framework for Approximate Matching Under Bounded Edits with Applications to Sequence Analysis

Sharma V. Thankachan¹(✉), Chaitanya Aluru², Sriram P. Chockalingam³,
and Srinivas Aluru^{3,4}

¹ Department of Computer Science, University of Central Florida, Orlando, FL, USA
sharma.thankachan@ucf.edu

² Department of Computer Science, Princeton University, Princeton, NJ, USA
caluru@princeton.edu

³ School of Computational Science and Engineering, Georgia Institute of Technology,
Atlanta, GA, USA
srirampc@gatech.edu, aluru@cc.gatech.edu

⁴ Institute for Data Engineering and Science, Georgia Institute of Technology,
Atlanta, GA, USA

Abstract. We present a novel algorithmic framework for solving approximate sequence matching problems that permit a bounded total number k of mismatches, insertions, and deletions. The core of the framework relies on transforming an approximate matching problem into a corresponding exact matching problem on suitably edited string suffixes, while carefully controlling the required number of such edited suffixes to enable the design of efficient algorithms. For a total input size of n , our framework limits the number of generated edited suffixes to no more than a factor of $O(\log^k n)$ of the input size (for any constant k), and restricts the algorithm to linear space usage by overlapping the generation and processing of edited suffixes. Our framework improves the best known upper bound of $n^2 k^{1.5} / 2^{\Omega(\sqrt{\log n/k})}$ for the classic k -edit longest common substring problem [Abboud, Williams, and Yu; SODA 2015] to yield the first strictly sub-quadratic time algorithm that runs in $O(n \log^k n)$ time and $O(n)$ space for any constant k . We present similar subquadratic time and linear space algorithms for (i) computing the alignment-free distance between two genomes based on the k -edit average common substring measure, (ii) mapping reads/read fragments to a reference genome while allowing up to k edits, and (iii) computing all-pair maximal k -edit common substrings (also, suffix/prefix overlaps), which has applications in clustering and assembly. We expect our algorithmic framework to be a broadly applicable theoretical tool, and may inspire the design of practical heuristics and software.

1 Introduction

Numerous problems related to exact sequence matching can be solved efficiently, often within optimal time and space bounds, typically using versatile string

data structures such as suffix trees and suffix arrays. However, variants of such sequence matching problems that permit a limited number of mismatches or edits (insertions/deletions/mismatches) are often challenging and many problems are still open. For example, the classic problem of finding the longest common substring (LCS¹) between a pair of sequences is easily solvable in optimal *linear time* using suffix trees, a solution that dates back to the 70's [31]. However, when the sequences contain (an unbounded number of) wild-card characters, an $n^{2-o(1)}$ time conditional lower bound, based on the Strong Exponential Time Hypothesis (SETH), comes into play [2]. As for the k -edit LCS problem², the best known result is only slightly better than a straightforward dynamic programming solution. Specifically, the run time is $n^2 k^{1.5} / 2^{\Omega(\sqrt{\log n/k})}$ and the algorithm is randomized [1].

In recent times, there is renewed interest in approximate sequence matching problems due to their wide applicability in computational biology. Many fundamental problems between evolutionarily related genomes, or components to the solutions thereof, can be cast as edit distance problems, with bounded versions of significant practical interest. Short read sequencers sport low error rates, typically <1–3% of sequence length, which is within a few hundred bases. Many problems in relating such reads to each other, or to the source genomes they originate from, can be effectively modeled as bounded edit distance problems. While many such problems can be solved efficiently in practice via heuristics, their worst-case run times are often the same as alignment-based methods that allow unconstrained edit distance. Thus, an algorithmic framework for approximate sequence matching that can lead to the design of strictly subquadratic time algorithms for such problems is of significant theoretical and practical interest.

Our Contributions and Relation to Prior Work

In this work, we focus on multiple approximate sequencing matching problems under a bounded number k of edits. We expect k to be a small constant in practice. Our algorithms work for arbitrary values of k , but they are designed to be superior in both asymptotic and practical runtimes for small values of k . We first develop a novel algorithmic framework that is potentially applicable to a broad class of problems, including the four problems solved in this paper. The core of the framework is a transformation by which an approximate matching problem on exact suffixes can be converted into an exact matching counterpart on approximate suffixes, specifically, suffixes with at most k edits. The number of such *k-edited suffixes* generated is constrained to a *polylog* factor of the input size, through a non-trivial application of Sleator and Tarjan's classic heavy path tree decomposition technique [26]. As a result, the framework yields algorithms with runtime behavior of $O(n \text{ polylog}(n))$ while consuming only linear $O(n)$ space, for a total input of size n , marking a significant improvement over current

¹ In this paper, we use LCS to denote the longest common substring. Note that LCS is frequently used in literature to refer to the longest common subsequence instead.

² Find the longest substring of a sequence that matches with a substring of another sequence, allowing $\leq k$ edits.

worst-case runtimes that are quadratic or near quadratic. Using this framework, we propose asymptotically faster algorithms for three well known and widely applicable problems in biological sequence analysis, and derive the first strictly sub-quadratic time algorithm for the k -edit LCS problem as a corollary to one of these. As will become evident later, the design of appropriate k -edited suffixes and algorithms for processing them are specific to the problem at hand, leading to a rich algorithmic framework for tackling additional approximate sequence matching problems.

Our first result concerns alignment-free genomic distance based on the average common substring (ACS) measure, proposed by Burstein *et al.* [9]. The ACS between genomes X and Y is:

$$\text{ACS}(X, Y) = \frac{1}{|X|} \sum_{i=1}^{|X|} L[i], \text{ where } L[i] = \max_j |\text{LCP}(X_i, Y_j)|$$

Here X_i is the i -th longest suffix of X and LCP denotes the longest common prefix. The distance metric based on ACS is defined as

$$\text{Dist}(X, Y) = \frac{1}{2} \left(\frac{\log |Y|}{\text{ACS}(X, Y)} + \frac{\log |X|}{\text{ACS}(Y, X)} \right) - \frac{1}{2} \left(\frac{\log |X|}{\text{ACS}(X, X)} + \frac{\log |Y|}{\text{ACS}(Y, Y)} \right).$$

Since its introduction, ACS has proven to be useful in multiple applications including phylogeny reconstruction [4, 6, 10, 12, 13, 15]. It was later observed that its approximate variants, k -mismatch and k -edit ACS, that are based on permitting k mismatches (or k edits, respectively) in the LCP computation, more accurately model genome evolution and lead to higher quality phylogenetic trees [18]. The ACS computation using exact substring composition, as described above, is straightforward to compute in linear time using suffix trees [9]. The k -mismatch ACS and the k -edit ACS can be computed by a trivial $O(n^2k)$ dynamic programming algorithm, which is prohibitively expensive for large genomes. Leimeister and Morgenstern [18] proposed algorithms that heuristically estimate k -mismatch ACS. Apostolico *et al.* [5] were the first to break the $O(n^2k)$ bound for an exact solution by proposing an $O(n^2/\log n)$ run time algorithm. However, the first strictly sub-quadratic algorithms are by Thankachan *et al.* [3, 27], that run in $O(n \log^k n)$ time. Also see [22, 24, 28, 29].

To date, there is no non-trivial solution for the k -edit ACS, beyond the straightforward $O(n^2k)$ algorithm. Unfortunately, the previous techniques for k -mismatch ACS do not easily extend to k -edit ACS. In comparison to mismatches, insertions and deletions are much harder to account for as they introduce a combinatorially larger number of possibilities (3^k ways of making modifications at k given locations) and also alter sequence lengths. Using the algorithmic framework presented in this paper, we present the first *strictly sub-quadratic* time algorithms for both k -edit ACS and k -edit LCS, that run in $O(n \log^k n)$ time and $O(n)$ space for any constant k^3 .

³ Throughout the analysis, we treat k as a constant for brevity. However, with a tighter analysis (deferred to full version), we can bound the time and space by $O(n(c \log n)^k/k!)$ and $O(c^k n)$, respectively for a constant c without making any such assumption on the value of k .

Theorem 1. *Given two sequences X and Y of n characters in total and a constant k , we can compute $\forall i, L[i] = \max_j |\text{LCP}_k(X_i, Y_j)|$ in $O(n \log^k n)$ time using $O(n)$ space. Here $|\text{LCP}_k(X_i, Y_j)|$ is the length of the longest common prefix of X_i and Y_j after allowing $\leq k$ edits.*

In addition, we provide sub-quadratic algorithms for the following problems. Note that the size of the alphabet set is $O(1)$ in all these applications, however we make no such assumptions in the complexity analysis.

- **Read mapping:** A collection of m reads of length ℓ each can be mapped to a reference genome G while permitting at most k edits per read in $O((n + \text{occ}) \log^k n)$ time using $O(n)$ space for any constant k . Here $n = |G| + m\ell$ is the input size and occ is the output size.
- **All-pair Maximal k -edit Common Substrings:** Given a collection of m reads of total length n , all pairwise k -edit maximal common substrings of length $\geq \tau$ can be computed in $O((n + \text{occ}) \log^k n)$ time using $O(n)$ space for any constant k . Here occ is the output size.
- **All-pair Maximal k -edit suffix/prefix overlaps:** Given a collection of m reads of total length n and a length threshold τ , all pairwise k -edit maximal suffix/prefix overlaps of length $\geq \tau$ can be computed in $O((n + \text{occ}) \log^k n)$ time using $O(n)$ space for any constant k . Here occ is the output size.

All of these are widely studied problems with excellent heuristic solutions and software availability. The read mapping problem is typically solved using seed-and-extend heuristics with exact matching or spaced seeds computed using a pre-built index of the genome such as BWT or FM-index (e.g. [17, 19, 21]; see [20] for a survey). Similarly, the other two problems are also solved through seed-and-extend type filtering solutions such as suffix filtering [16, 30], spaced seeds filtering [8], and substring filtering [25]. Our goal is to present asymptotically efficient and sub-quadratic worst-case run-time algorithms for these commonly solved problems to improve upon their upper bounds. We remark that the algorithms presented here can also be used in conjunction with any existing seed-based heuristics by permitting seeds with bounded edit distance.

Roadmap. In Sect. 2, we present an overview of our framework and the key results, which are instrumental in achieving the above claimed worst-case run times. The proofs of the key results of our framework are described in detail in Sect. 3. We complete the proof of Theorem 1 in Sect. 4. In Sect. 5, we present our solutions to the other problems listed.

2 Our Algorithmic Framework

Our approximate sequence matching framework takes a collection of two or more sequences and a constant k as input. Then, a controlled number of changes (edits) are applied to the suffixes of all input sequences, so that an approximate sequence matching task over the input can now be transformed to an equivalent exact prefix matching over the newly generated edited-suffixes. We illustrate our

framework with a collection of two input sequences (X and Y of total length n). The framework relies on a Generalized Suffix Tree (GST), a compact trie representation of all suffixes of all input sequences. It takes $O(n)$ space for storage and $O(n)$ time for construction [23, 31]. For any two suffixes X_i and Y_j , we can compute $|\text{LCP}_0(X_i, Y_j)| = |\text{LCP}(X_i, Y_j)| = z$ in constant time using GST and $|\text{LCP}_k(X_i, Y_j)|$ for any $k > 0$ in $O(3^k)$ time via the following recursion:

$$|\text{LCP}_k(X_i, Y_j)| = z + \max \begin{cases} 1 + |\text{LCP}_{k-1}(X_{i+z+1}, Y_{j+z+1})| & \text{(substitution)} \\ |\text{LCP}_{k-1}(X_{i+z+1}, Y_{j+z})| & \text{(deletion in } X_i) \\ |\text{LCP}_{k-1}(X_{i+z}, Y_{j+z+1})| & \text{(deletion in } Y_j) \end{cases}$$

Observe that while computing LCP_k , a substitution (in at least one suffix) is equivalent to deletions in both suffixes at the same location. For example, $X_i = \text{AATCGGT}..$ and $Y_j = \text{AATGGTT}..$ disagree at the 4th position. To make them agree more, we can either delete the 4th character from both suffixes, or change the 4th character in at least one suffix to match the 4th character of the other. Also, deletion in X_i (respectively, Y_j) is equivalent to an appropriate insertion in Y_j (respectively, X_i). Therefore, in general we have many possible (equivalent) ways of correcting the first k disagreements between X_i and Y_j . Note that the length of the resulting LCP_k may differ (slightly) as per our choice within the equivalent cases. However, the framework we propose exploits the fact that many of the equivalent cases will lead to the correct solution, and makes a suitable fixed choice.

Overview. A suffix after applying $\leq k$ edits is called a k -**edited suffix**. Let X'_i and Y'_j be k -edited suffixes derived from X_i and Y_j , respectively. Then, the value of $|\text{LCP}(X'_i, Y'_j)|$ can range anywhere between 0 and $|\text{LCP}_{2k}(X_i, Y_j)|$. However, if the modifications turn **exactly** the first k disagreeing positions into agreements, then $|\text{LCP}(X'_i, Y'_j)|$ is precisely $|\text{LCP}_k(X_i, Y_j)|$. A **set of two** such edited suffixes is called an $(i, j)_k$ -**maxpair**. We call a collection of k -edited suffixes an order- k universe (denoted by U_k) if for all (i, j) pairs, $\exists(i, j)_k\text{-maxpair} \subseteq U_k$. Note that U_0 is simply the set of all suffixes of X and Y . Trivially, there exists an order- k universe of size $\binom{n}{2}$. However, the core of our framework is a meticulous construction of an order k universe of size $O(n \log^k n)$, based on the *heavy path decomposition* strategy by Sleator and Tarjan [26] as in Cole *et al.* [11]. Various approximate sequence matching problems can then be solved via processing U_k in linear or near-linear time.

Representation of Edited Suffixes. Clearly, it is cumbersome to keep track of all edits applied on suffixes during the creation of edited-suffixes. However, we have the following crucial observation: *for each edited suffix, we do not need to keep track of all edits, but only substitutions and the total number of insertions and deletions*. Specifically, let X'_i be a k -edited suffix obtained via a combination of insertions, deletions and substitutions on X_i . Then, X'_i can be simply represented as a concatenation of a combination of $O(k)$ sub-strings of X and characters in the alphabet set, along with the following two satellite information.

- $\delta(X'_i)$: **number** of insertions and deletions made to transform X_i to X'_i .
- $\Delta(X'_i)$: **set** of positions in X'_i corresponding to substitutions in X_i .

Example: Let $X_i = CATCATCATCAT$. We consider the following edits simultaneously on X_i : delete the 2nd and 10th character, change the 4th character to T and the 9th character to A, and insert G after position 6. Then, $X'_i = CTTATGCA\textit{\underline{A}}AT$, $\delta(X'_i) = 3$ and $\Delta(X'_i) = \{3, 9\}$.

Lemma 1. *Let X'_i (respectively, Y'_j) be obtained via at most k edits on X_i (respectively, Y_j). Then, the value of $|\text{LCP}(X'_i, Y'_j)|$ can range anywhere between 0 and $|\text{LCP}_{2k}(X_i, Y_j)|$. However, if we impose the following condition, then $|\text{LCP}(X'_i, Y'_j)|$ is at most $|\text{LCP}_k(X_i, Y_j)|$.*

$$|\Delta(X'_i) \cup \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j) \leq k.$$

Proof. If we allow k edits on each suffix, we can correct at most $2k$ disagreements. However, the condition limits the total number of insertions/deletions and distinct substitution positions. □

We now define the notion of $(i, j)_k$ -maxpair in a formal way.

Definition 1. *Let X'_i be a k -edited suffix derived from X_i and Y'_j be a k -edited suffix derived from Y_j . Then, we call the set $\{X'_i, Y'_j\}$ an $(i, j)_k$ -maxpair iff*

$$|\text{LCP}(X'_i, Y'_j)| = |\text{LCP}_k(X_i, Y_j)| \text{ and } |\Delta(X'_i) \cup \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j) \leq k.$$

Lemma 2. *Given two k -edited suffixes, we can compute the length of their longest common prefix (hence their lexicographic order) in $O(k)$ time via $O(k)$ number of $|\text{LCP}|$ queries on the GST.*

3 Details of the Construction of U_k

We show how to construct the universe U_k in small parts (in linear work space). The **parts** of U_k , denoted by $\{\mathcal{P}_1^k, \mathcal{P}_2^k, \mathcal{P}_3^k, \dots\}$ are its subsets (not necessary disjoint) such that the following properties are ensured.

1. $\max_f |\mathcal{P}_f^k| = O(n)$
2. $\sum_f |\mathcal{P}_f^k| = O(n \log^k n)$
3. for any (i, j) , $\exists f$ such that a two-element subset of \mathcal{P}_f^k is an $(i, j)_k$ -maxpair

The construction procedure is recursive. We first construct U_0 , then U_1 from U_0 and so on. The base case, i.e., an order 0 universe U_0 has exactly one part, the set of all suffixes of X and Y . We now proceed to the inductive step, where we assume the availability of order- h universe U_h (specifically, its parts $\mathcal{P}_1^h, \mathcal{P}_2^h, \dots$) for an $h \geq 0$ and the task is to obtain the parts $\mathcal{P}_1^{h+1}, \mathcal{P}_2^{h+1}, \dots$ of U_{h+1} . To do so, we apply the following steps on each \mathcal{P}_f^h . We describe the procedure first and prove its correctness later.

1. Let $m = |\mathcal{P}_f^h|$ and \mathcal{T} be a compact trie of all h -edited suffixes in \mathcal{P}_f^h . Notice that \mathcal{T} is GST when $h = 0$. Classify the nodes in \mathcal{T} into **light** or **heavy**: the root is always light and any other node is heavy, if it is the *heaviest child*⁴ of its parent. Furthermore, a maximal downward path starting from a light node where all other nodes on the path are heavy is called a **heavy path**. A key property is that the *number of heavy paths that intersect any root to leaf path is $\leq \log m$* [11, 26]. Equivalently, the number of light nodes on any root to leaf path is $\leq \log m$. Therefore the sum of subtree sizes of all light nodes in \mathcal{T} is $\leq m \log m$, because each leaf contributes to at most $\log m$ light rooted subtree.
2. Corresponding to each internal *light* node u in \mathcal{T} , there will be a **part**, say \mathcal{P}_t^{h+1} . The steps involved in its construction are as follows. Let Q be the set of h -edited suffixes corresponding to the leaves in the subtree of u , α be the h -edited suffix corresponding to the particular leaf on the heavy path through u . Then,

$$\mathcal{P}_t^{h+1} = \{\alpha\} \cup_{\beta \in Q, \beta \neq \alpha} \{\beta, \beta^I, \beta^D, \beta^S\}$$

Here, β^I, β^D and β^S are $(h+1)$ -edited suffixes, obtained by performing exactly one edit on β w.r.t. α as follows: Let $z = |\text{LCP}(\alpha, \beta)|$ and σ be the $(z+1)$ th character of α , then

- β^I is obtained by *inserting* the character σ in β after the z th character.
- β^D is obtained by *deleting* the $(z+1)$ th character of β .
- β^S is obtained by *substituting* the $(z+1)$ th character of β by σ .

See Fig. 1 for an illustration. We now prove that the parts created in the above manner satisfy the desired properties.

3.1 Correctness Proof (via Mathematical Induction)

All three properties hold true for $k = 0$ (base case). Assuming they are true for all values of k up to h , we now prove it for $h+1$. From our construction procedure, $|\mathcal{P}_t^{h+1}| = 1 + 4(|Q| - 1) < 4|\mathcal{P}_f^h| = 4m$. Therefore, the maximum size of a part can be bounded by $\max_t |\mathcal{P}_t^{h+1}| < 4 \max_f |\mathcal{P}_f^h| = O(n)$. The total size of all pairs derived from \mathcal{P}_f^h is $4 \sum_{u \text{ is light}} \text{subtree-size}(u) \leq 4m \log m$. Therefore, the total size of all parts in U_{h+1} is

$$\sum_t |\mathcal{P}_t^{h+1}| \leq 4 \sum_f |\mathcal{P}_f^h| \log |\mathcal{P}_f^h| < 4 \left(\sum_f |\mathcal{P}_f^h| \right) \left(\log \sum_f |\mathcal{P}_f^h| \right) = O(n \log^{h+1} n)$$

Next we prove the existence of an $(i, j)_{h+1}$ -maxpair in at least one part, say \mathcal{P}_t^{h+1} . Without loss of generality, assume $\{X'_i, Y'_j\}$ is an $(i, j)_h$ -maxpair and is a subset of \mathcal{P}_f^h . Then,

$$|\text{LCP}(X'_i, Y'_j)| = |\text{LCP}_h(X_i, Y_j)| \quad \text{and} \quad |\Delta(X'_i) \cup \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j) \leq h$$

⁴ The child with the largest number of leaves in its subtree (ties broken arbitrarily) among its siblings.

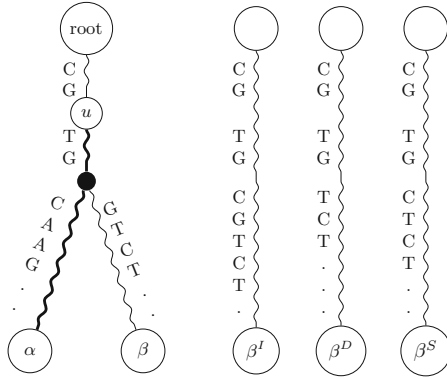


Fig. 1. Illustrates an edit operation along a suffix β , at the point where it diverges from a heavy path (shown as a thick wavy line). For insertion (β^I) and substitution (β^S), the modification is made to conform to the next character along the heavy path.

Let w be the lowest common ancestor of the leaves corresponding to X'_i and Y'_j in the trie \mathcal{T} , $l = |\text{LCP}(X'_i, Y'_j)|$, σ be the leading character on the outgoing edge from w towards its heavy child, and \mathcal{P}_t^{h+1} be the part created w.r.t. the heavy path through w . We now prove there exists an $(i, j)_{h+1}\text{-maxpair} \subseteq \mathcal{P}_t^{h+1}$. We have the following cases.

Case 1: At w , both X'_i and Y'_j diverge from the heavy path through w . Then the following edited suffixes, in addition to X'_i and Y'_j , are in \mathcal{P}_f^{h+1} . Let

- X''_i be the edited suffix obtained by deleting the $(l + 1)$ th character of X'_i .
- Y''_j be the edited suffix obtained by deleting the $(l + 1)$ th character of Y'_j .
- X'''_i be the edited suffix obtained by substituting the $(l + 1)$ th character of X'_i by σ .
- Y'''_j be the edited suffix obtained by substituting the $(l + 1)$ th character of Y'_j by σ .

It can be easily verified that one of the following subsets of \mathcal{P}_t^{h+1} is an $(i, j)_{h+1}\text{-maxpair}$: $\{X'_i, Y'_j\}, \{X''_i, Y''_j\}, \{X'''_i, Y'''_j\}$.

Case 2: At w , exactly one among X'_i and Y'_j diverges from the heavy path. Without loss of generality, assume the diverging suffix is X'_i . Then the following edited suffixes, in addition to X'_i and Y'_j , are in \mathcal{P}_f^{h+1} . Let

- X''_i be the edited suffix obtained by deleting the $(l + 1)$ th character of X'_i .
- X'''_i be the edited suffix obtained by substituting the $(l + 1)$ th character of X'_i by σ .
- X''''_i be the edited suffix obtained by inserting σ in X'_i after l characters.

Here also, it can be easily verified that one of the following subset of \mathcal{P}_t^{h+1} is an $(i, j)_{h+1}\text{-maxpair}$: $\{X''_i, Y'_j\}, \{X'''_i, Y'_j\}, \{X''''_i, Y'_j\}$. This completes the correctness proof.

3.2 Time and Space Complexity Analysis

First, we consider the recursive step of creating parts out of \mathcal{P}_f^h . The trie \mathcal{T} can be constructed in $O(m \log m)$ time (recall $m = |\mathcal{P}_f^h|$) with the following steps.

1. First, sort the edited suffixes in \mathcal{P}_f^h in time $O(m \log m)$ via merge sorting. Note that any two k -edited suffixes can be compared in $O(k)$ time (refer to Lemma 2).
2. Then, compute the LCP between every consecutive pair of edited suffixes in the sorted list and build the trie \mathcal{T} using standard techniques from the suffix tree construction algorithms [14]. This step takes only $O(m)$ time.

Note that the part corresponding to each light node u can be obtained in time proportional to the subtree size of u . Therefore, the time for deriving parts from \mathcal{P}_f^h is $O(m \log m)$. In other words, parts of U_{h+1} can be obtained from parts of U_h in $O(\log n \sum_f |\mathcal{P}_f^h|)$ time for $h = 0, 1, 2, \dots, k-1$. Total time is $\log n \sum_{h=0}^{k-1} |\mathcal{P}_f^h| = O(n \log^k n)$.

The parts can be created (and processed) one at a time by keeping exactly one partition in each U_h for $h = 0, 1, 2, \dots, k$. Therefore, the working space is $\sum_{h=0}^k \max_t |\mathcal{P}_t^h| = O(n)$.

Lemma 3. *The universe U_k can be created in parts in $O(n \log^k n)$ time using $O(n)$ space.*

3.3 Obtaining the Parts of U_k with Its Elements Sorted

We now present a more careful implementation of the above steps, so that the parts can be generated with their elements in sorted order without incurring additional comparison sorting costs. Specifically, we show how to process a light node u in \mathcal{T} (the trie over all edited suffixes in \mathcal{P}_f^h) and construct the corresponding part \mathcal{P}_t^{h+1} in U_{h+1} with its elements sorted. We use the classic result that two sorted lists of sizes p and q ($q \leq p$) in the form of balanced binary search trees (BSTs) can be merged using $O(q \log(p/q))$ comparisons [7]. Throughout the execution of our algorithm, we maintain edited suffixes in the form of a BST. Key steps are below.

1. Initialize BST with exactly one element α .
2. Visit the heavy internal nodes on the heavy path through u in a bottom up fashion. For each light child w of a heavy node v on the path (let l be the string depth of v), merge BST with BST_w , BST_w^I , BST_w^D and BST_w^S . Here,
 - BST_w is the set of all strings corresponding to the leaves in the subtree of w .
 - BST_w^I is BST_w after inserting the character $\alpha[l+1]$ after l th position of all strings in it.
 - BST_w^D is BST_w after deleting the $(l+1)$ th character of all strings in it.

- BST_w^S is BST_w after replacing the $(l + 1)$ th character by $\alpha[l + 1]$ for all its strings.

Note that BST_w can be created from \mathcal{T} in time linear to its size. Since the LCP of any two strings in BST_w is at least $(l + 1)$, we can generate BST_w^I , BST_w^D and BST_w^S also in time linear to their size. Therefore, the merging can be performed in time $O(\text{size}(w) \log(\text{size}(v)/\text{size}(w)))$ via fast merging.

The correctness is ensured as we are implementing the same algorithm described earlier. The time for processing all light nodes in \mathcal{T} is the sum of $\text{size}(\cdot) \times \log(\text{size}(\text{parent}(\cdot))/\text{size}(\cdot))$ over all light nodes. This is the same as the sum of $\log(\text{size}(\text{parent}(\cdot))/\text{size}(\cdot))$ over all light ancestors of all leaves. However, sum of $\log(\text{size}(\text{parent}(\cdot))/\text{size}(\cdot))$ over all nodes on any root to leaf path is $\log(\text{size}(\text{root}))$. In summary, we have the following.

Lemma 4. *We can generate U_k with its parts sorted in $O(n \log^k n)$ time using $O(n)$ space.*

4 Our Algorithm for Computing the Array L

We can compute $\forall i, L[i] = \max_j |\text{LCP}_k(X_i, Y_j)|$ with the following procedure. First, initialize all entries in array L to 0 and then, process each part \mathcal{P}_f^k one after another as follows:

$$\begin{aligned} &\forall X'_i, Y'_j \in \mathcal{P}_f^k \text{ s.t. } |\Delta(X'_i) \cup \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j) \leq k, \\ &\quad \text{update } L[i] \leftarrow \max\{L[i], |\text{LCP}(X'_i, Y'_j)|\} \end{aligned}$$

After processing all the parts of U_k , we have $\max_j |\text{LCP}_k(X_i, Y_j)| = L[i]$ for all values of i . Correctness follows from the fact that at some point during the execution of the algorithm, we will process a pair X_i^*, Y_d^* corresponding an (i, d) -maxpair with $d = \arg \max_j |\text{LCP}_k(X_i, Y_j)|$ and update $L[i] \leftarrow |\text{LCP}_k(X_i, Y_d)|$. However, we cannot afford to examine all the pairs.

Our Strategy. $\forall h, t \in [0, k]$ and set ϕ , generate all non-empty sets $S(h, t, \phi)$ from \mathcal{P}_f^k , such that $S(h, t, \phi) =$

$$\{X'_i \mid \phi \subseteq \Delta(X'_i) \text{ and } |\Delta(X'_i)| + \delta(X'_i) = h\} \cup \{Y'_j \mid \phi \subseteq \Delta(Y'_j) \text{ and } |\Delta(Y'_j)| + \delta(Y'_j) = t\}$$

Observe that $\forall X'_i, Y'_j \in S(h, t, \phi)$, $|\Delta(X'_i) \cup \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j)$

$$\begin{aligned} &= |\Delta(X'_i)| + |\Delta(Y'_j)| - |\Delta(X'_i) \cap \Delta(Y'_j)| + \delta(X'_i) + \delta(Y'_j) \\ &= h + t - |\Delta(X'_i) \cap \Delta(Y'_j)| \\ &\leq h + t - |\phi| \end{aligned}$$

This in turn implies that $|\text{LCP}(X'_i, Y'_j)| \leq |\text{LCP}_{h+t-|\phi|}(X_i, Y_j)|$. Therefore,

$$\forall X'_i, Y'_j \in S(h, t, \phi) \text{ with } h + t - |\phi| \leq k, |\text{LCP}(X'_i, Y'_j)| \leq |\text{LCP}_k(X_i, Y_j)|$$

Additionally, $\forall (i, j)$ pairs, there exists an $(i, j)_k$ -maxpair, say $\{X''_i, Y''_j\}$ and an f , such that $X''_i, Y''_j \in \mathcal{P}_f^k$. In other words, there exists a non-empty set $S(a, b, \mu)$, such that $X''_i, Y''_j \in S(a, b, \mu)$. Specifically, $a = |\Delta(X''_i)| + \delta(X''_i)$, $b = |\Delta(Y''_j)| + \delta(Y''_j)$ and $\mu =$

$\Delta(X''_i) \cap \Delta(Y''_j)$. Since $\{X''_i, Y''_j\}$ is an $(i, j)_k$ -maxpair, $|\text{LCP}(X''_i, Y''_j)| = |\text{LCP}_k(X_i, Y_j)|$ and $a + b - |\mu| \leq k$. Therefore,

$$|\text{LCP}_k(X_i, Y_j)| = \max\{|\text{LCP}(X'_i, Y'_j)| \mid X'_i, Y'_j \in S(h, t, \phi) \text{ and } (h + t - |\phi| \leq k)\}$$

$$L[i] = \max_j \{|\text{LCP}(X'_i, Y'_j)| \mid X'_i, Y'_j \in S(h, t, \phi) \text{ and } (h + t - |\phi| \leq k)\}$$

Note that there is no $|\text{LCP}_k(\cdot, \cdot)|$ in the above equation. Equivalently, we have a new definition for $L[\cdot]$ using *exact matching over k -edited suffixes*. Therefore, the computation of L is straightforward.

Proposed Algorithm. Initialize $L[i] \leftarrow 0, \forall i$. Then, $\forall h, t \in [0, k]$ and set ϕ with $h + t - |\phi| \leq k$, process $S(h, t, \phi)$ as follows: sort all of its strings, and visit the strings in both ascending and descending order. For each X'_i visited, update $L[i] \leftarrow \max\{L[i], |\text{LCP}(X'_i, Y'_j)|\}$, where Y'_j is the last visited k -edited suffix of Y . Correctness is immediate from the above discussions.

Space and Time Analysis. Since we process the parts \mathcal{P}_f^k one after another, space is $O(n)$. W.r.t. time complexity, note that each $S(\cdot, \cdot, \cdot)$ can be processed in time linear plus the time for sorting its strings, which is $O(|S(\cdot, \cdot, \cdot)| \log(|S(\cdot, \cdot, \cdot)|))$ using Lemma 2. The sum of sizes of all $S(\cdot, \cdot, \cdot)$ generated from a particular \mathcal{P}_f^k is at most $k \times 2^k \times |\mathcal{P}_f^k|$ i.e., $\sum |S(\cdot, \cdot, \cdot)| = O(n \log^k n)$. Total time is $\sum |S(\cdot, \cdot, \cdot)| \log(|S(\cdot, \cdot, \cdot)|) = O(n \log^{k+1} n)$.

To shave off an additional $\log n$ factor from the time complexity, we replace the merge sorting by integer sorting. Specifically, we generate all \mathcal{P}_f^k 's with their elements sorted using Lemma 4. We then process \mathcal{P}_f^k s after replacing each edited suffix within \mathcal{P}_f^k by its lexicographic rank in \mathcal{P}_f^k . Essentially, we replace all string comparison tasks by integer comparison. Therefore, the main task now is the sorting of several sets of integers of total size $O(n \log^k n)$ and maximum size $O(n)$. On sets of size $\Theta(n)$, we employ counting sort. To sort smaller sets, we combine several of them up to a total size of $\Theta(n)$. Then, a counting sort is performed, followed by a stable sort with the id associated with the set in which each integer belongs to as the key. By scanning the output in linear time, we can segregate the individual sorted lists. The time in both cases is constant per element. By combining this with Lemma 4, we obtain the result in Theorem 1.

5 Solving Approximate Sequence Matching Problems

5.1 Computing the k -edit Average Common Substring

The computation of ACS_k from L is straightforward. We now demonstrate the applicability of our algorithmic framework to three other important problems. The general strategy is to begin with an order-0 universe U_0 with one part: the set of all suffixes of all input sequences. Then create U_k , in parts and process them one by one, using problem specific steps. In all three cases, the correctness (deferred to full version) can be obtained via straightforward adaptations of the correctness proof of Theorem 1.

5.2 Our Algorithm for the Mapping Problem

Let $\{R_1, R_2, \dots, R_m\}$ be the set of input reads and G be the reference genome. Our task is to report all (i, j) pairs, s.t. the edit distance between R_i and $G[j..(j + \ell - 1)]$ is $\leq k$, where ℓ is the read length. We use R'_i (resp., G'_j) for a k -edited copy of R_i (resp., G_j). Let $S(h, t, \phi)$ w.r.t. \mathcal{P}_f^k be $\{R'_i \in \mathcal{P}_f^k \mid |\Delta(R'_i)| + \delta(R'_i) = h \text{ and } \phi \subseteq \Delta(R'_i)\} \cup \{G'_j \in \mathcal{P}_f^k \mid |\Delta(G'_j)| + \delta(G'_j) = t \text{ and } \phi \subseteq \Delta(G'_j)\}$. Then, $\forall h, t \in [0, k]$ and set ϕ with $h + t - |\phi| \leq k$, process all $S(h, t, \phi)$ as follows: $\forall R'_i, G'_j \in S(h, t, \phi)$ s.t. $|\text{LCP}(R'_i, G'_j)| \geq \ell$, report (i, j) . The following correctness argument can be easily verified: we report a pair (i, j) iff it is a valid output.

The processing of an $S(\cdot, \cdot, \cdot)$ is now an exact matching task. It can be implemented in time linear to its size and the number of pairs reported. Therefore, time over all $S(\cdot, \cdot, \cdot)$ is $O(n \log^k n)$ plus the total number of pairs reported. Note that our algorithm might report the same pair multiple times, but not more than $O(\log^k n)$ times, because for any (i, j) pair, the number of parts containing an R'_i and a G'_j is $O(\log^k n)$ (follows from our construction). Therefore total time is $O((n + occ) \log^k n)$.

5.3 All-Pair Maximal k -edit Common Substrings

Let $\{R_1, R_2, \dots, R_m\}$ be the set of input reads. Let $R_{i,x}$ denotes the x th longest suffix of R_i and let $R'_{i,x}$ denotes a k -edited copy of $R_{i,x}$. Our task is to report all tuples (i, x, j, y) , s.t. $|\text{LCP}_k(R_{i,x}, R_{j,y})| \geq \tau$, $i \neq j$ and $R_i[x - 1] \neq R_j[y - 1]$. Let $S(h, t, \phi)$ w.r.t. a part \mathcal{P}_f^k is the union of the following two sets.

$$\begin{aligned} &\{R'_{i,x} \in \mathcal{P}_f^k \mid |\Delta(R'_{i,x})| + \delta(R'_{i,x}) = h \text{ and } \phi \subseteq \Delta(R'_{i,x})\} \\ &\{R'_{j,y} \in \mathcal{P}_f^k \mid |\Delta(R'_{j,y})| + \delta(R'_{j,y}) = t \text{ and } \phi \subseteq \Delta(R'_{j,y})\} \end{aligned}$$

Then, $\forall h, t \in [0, k]$ and set ϕ with $h + t - |\phi| \leq k$, process $S(h, t, \phi)$ as follows: $\forall R'_{i,x}, R'_{j,y} \in S(h, t, \phi)$ s.t. $|\text{LCP}(R'_{i,x}, R'_{j,y})| \geq \tau$, $|\Delta(R'_{i,x})| + \delta(R'_{i,x}) = h$, $|\Delta(R'_{j,y})| + \delta(R'_{j,y}) = t$, $R_i[x - 1] \neq R_j[y - 1]$ and $i \neq j$, report (i, x, j, y) . This (exact matching) task can be easily implemented in time linear to $|S(h, t, \phi)|$ and the number of tuples generated using standard techniques (details deferred to full version). Also, we report a tuple iff it is a valid output and we report one only $O(\log^k n)$ times. Hence the total run time is $O((n + occ) \log^k n)$.

5.4 All-Pair Maximal k -edit Suffix/Prefix Overlaps

Borrowing from the terminologies defined in Sect. 5.3, the task here is to report all tuples (i, j, y) , s.t. $i \neq j$ and $|\text{LCP}_k(R_i, R_{j,y})| \geq (\ell - y + 1) \geq \tau$. To do so, we process all $S(h, t, \phi)$ with $h + t - |\phi| \leq k$ as follows: $\forall R'_i, R'_{j,y} \in S(h, t, \phi)$ s.t. $|\text{LCP}(R'_i, R'_{j,y})| \geq \ell - y + 1 \geq \tau$, $|\Delta(R'_i)| + \delta(R'_i) = h$, $|\Delta(R'_{j,y})| + \delta(R'_{j,y}) = t$ and $i \neq j$, report (i, j, y) . Again, this is an exact matching task, which can be easily implemented in time linear to $|S(h, t, \phi)|$ and the number of tuples generated using standard techniques (details deferred to full version). Also, we report a tuple iff it is a valid output and we report one only $O(\log^k n)$ times across all $S(\cdot, \cdot, \cdot)$'s, yielding $O((n + occ) \log^k n)$ total time.

Acknowledgments. This research is supported in part by the U.S. National Science Foundation under CCF-1704552 and CCF-1703489.

References

1. Abboud, A., Williams, R., Yu, H.: More applications of the polynomial method to algorithm design. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 218–230 (2015)
2. Abboud, A., Williams, V.V., Weimann, O.: Consequences of faster alignment of sequences. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 39–51. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43948-7_4
3. Aluru, S., Apostolico, A., Thankachan, S.V.: Efficient alignment free sequence comparison with bounded mismatches. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 1–12. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16706-0_1
4. Apostolico, A.: Maximal words in sequence comparisons based on subword composition. In: Elomaa, T., Mannila, H., Orponen, P. (eds.) Algorithms and Applications. LNCS, vol. 6060, pp. 34–44. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12476-1_2
5. Apostolico, A., Guerra, C., Landau, G.M., Pizzi, C.: Sequence similarity measures based on bounded hamming distance. *Theoret. Comput. Sci.* **638**, 76–90 (2016)
6. Bonham-Carter, O., Steele, J., Bastola, D.: Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Briefings Bioinform.* **15**(6), 890–905 (2013)
7. Brown, M.R., Tarjan, R.E.: A fast merging algorithm. *J. ACM* **26**(2), 211–226 (1979)
8. Burkhardt, S., Kärkkäinen, J.: Better filtering with gapped q-grams. *Fundam. Inform.* **56**(1–2), 51–70 (2003)
9. Burstein, D., Ulitsky, I., Tuller, T., Chor, B.: Information theoretic approaches to whole genome phylogenies. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2005. LNCS, vol. 3500, pp. 283–295. Springer, Heidelberg (2005). https://doi.org/10.1007/11415770_22
10. Chang, G., Wang, T.: Phylogenetic analysis of protein sequences based on distribution of length about common substring. *Protein J.* **30**(3), 167–172 (2011)
11. Cole, R., Gottlieb, L.-A., Lewenstein, M.: Dictionary matching and indexing with errors and don't cares. In: Proceedings of the 36th Annual ACM Symposium on Theory of computing (STOC), pp. 91–100. ACM (2004)
12. Comin, M., Verzotto, D.: Alignment-free phylogeny of whole genomes using underlying subwords. *Algorithms Mol. Biol.* **7**(1), 1 (2012)
13. Domazet-Lošo, M., Haubold, B.: Efficient estimation of pairwise distances between genomes. *Bioinformatics* **25**(24), 3221–3227 (2009)
14. Gusfield, D.: *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge (1997)
15. Guyon, F., Brochier-Armanet, C., Guénoche, A.: Comparison of alignment free string distances for complete genome phylogeny. *Adv. Data Anal. Classif.* **3**(2), 95–108 (2009)
16. Kucherov, G., Tsur, D.: Improved filters for the approximate suffix-prefix overlap problem. In: Moura, E., Crochemore, M. (eds.) SPIRE 2014. LNCS, vol. 8799, pp. 139–148. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11918-2_14
17. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**(4), 357–359 (2012)

18. Leimeister, C.-A., Morgenstern, B.: kmacs: the k-mismatch average common substring approach to alignment-free sequence comparison. *Bioinformatics* **30**(14), 2000–2008 (2014)
19. Li, H., Durbin, R.: Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics* **25**(14), 1754–1760 (2009)
20. Li, H., Homer, N.: A survey of sequence alignment algorithms for next-generation sequencing. *Briefings Bioinform.* **11**(5), 473–483 (2010)
21. Li, R., Yu, C., Li, Y., Lam, T.-W., Yiu, S.-M., Kristiansen, K., Wang, J.: SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* **25**(15), 1966–1967 (2009)
22. Manzini, G.: Longest common prefix with mismatches. In: Iliopoulos, C., Puglisi, S., Yilmaz, E. (eds.) SPIRE 2015. LNCS, vol. 9309, pp. 299–310. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23826-5_29
23. McCreight, E.M.: A space-economical suffix tree construction algorithm. *J. ACM (JACM)* **23**(2), 262–272 (1976)
24. Pizzi, C.: A filtering approach for alignment-free biosequences comparison with mismatches. In: Pop, M., Touzet, H. (eds.) WABI 2015. LNCS, vol. 9289, pp. 231–242. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48221-6_17
25. Simpson, J.T., Durbin, R.: Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* **22**(3), 549–556 (2012)
26. Sleator, D.D., Tarjan, R.E.: A data structure for dynamic trees. *J. Comput. Syst. Sci.* **26**(3), 362–391 (1983)
27. Thankachan, S.V., Apostolico, A., Aluru, S.: A provably efficient algorithm for the k-mismatch average common substring problem. *J. Comput. Biol.* **23**(6), 472–482 (2016)
28. Thankachan, S.V., Chockalingam, S.P., Liu, Y., Apostolico, A., Aluru, S.: ALFRED: a practical method for alignment-free distance computation. *J. Comput. Biol.* **23**(6), 452–460 (2016)
29. Thankachan, S.V., Chockalingam, S.P., Liu, Y., Krishnan, A., Aluru, S.: A greedy alignment-free distance estimator for phylogenetic inference. In: Proceedings of 5th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS) (2015)
30. Välimäki, N., Ladra, S., Mäkinen, V.: Approximate all-pairs suffix/prefix overlaps. *Inf. Comput.* **213**, 49–58 (2012)
31. Weiner, P.: Linear pattern matching algorithms. In: Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory (SWAT), pp. 1–11 (1973)



Accurate Reconstruction of Microbial Strains from Metagenomic Sequencing Using Representative Reference Genomes

Zhemín Zhou^(✉), Nina Luhmann^(✉), Nabil-Fareed Alikhan,
Christopher Quince, and Mark Achtman^(✉)

Warwick Medical School, University of Warwick, Coventry, UK
{zhemin.zhou,n.luhmann,m.achtman}@warwick.ac.uk

Abstract. Exploring the genetic diversity of microbes within the environment through metagenomic sequencing first requires classifying these reads into taxonomic groups. Current methods compare these sequencing data with existing biased and limited reference databases. Several recent evaluation studies demonstrate that current methods either lack sufficient sensitivity for species-level assignments or suffer from false positives, overestimating the number of species in the metagenome. Both are especially problematic for the identification of low-abundance microbial species, e.g. detecting pathogens in ancient metagenomic samples. We present a new method, SPARSE, which improves taxonomic assignments of metagenomic reads. SPARSE balances existing biased reference databases by grouping reference genomes into similarity-based hierarchical clusters, implemented as an efficient incremental data structure. SPARSE assigns reads to these clusters using a probabilistic model, which specifically penalizes non-specific mappings of reads from unknown sources and hence reduces false-positive assignments. Our evaluation on simulated datasets from two recent evaluation studies demonstrated the improved precision of SPARSE in comparison to other methods for species-level classification. In a third simulation, our method successfully differentiated multiple co-existing *Escherichia coli* strains from the same sample. In real archaeological datasets, SPARSE identified ancient pathogens with $\leq 0.02\%$ abundance, consistent with published findings that required additional sequencing data. In these datasets, other methods either missed targeted pathogens or reported non-existent ones.

SPARSE and all evaluation scripts are available at <https://github.com/zheminzhou/SPARSE>.

1 Introduction

Shotgun metagenomics generates DNA sequences directly from environmental samples, revealing unculturable organisms in the community as well as those that can be isolated. The resulting data represents a pool of all species within a sample, thus raising the problem of identifying individual microbial species and their

relative abundance within these samples. Methods for such taxonomic assignment are either based on *de novo* assembly of the metagenomic reads, or take advantage of comparisons to existing *reference* genomes. Here we concentrate on the latter strategy, which relies on the diversity of genomes in ever-growing reference databases. This strategy has been instrumental in identifying many causative agents of ancient pandemics in reads obtained from archaeological samples by detecting genetic signatures of modern human pathogens [26].

Published methods for taxonomic assignment can be divided into two categories. *Taxonomic profilers* maintain a small set of curated genomic markers, which can be universal (e.g. used in MIDAS [16]) or clade-specific (e.g. used in MetaPhlan2 [24]). Metagenomic reads that align onto these genomic markers are used to extrapolate the taxonomic composition of the whole sample. These tools are usually computationally efficient with good precision. However, they also tend to show reduced resolution for species-level assignment [23], especially when a species has a low abundance in the sample and, hence, may have few reads mapping to a restricted set of markers.

Alternatively, *taxonomic bidders* compare metagenomic reads against reference genomes to achieve read-level taxonomic classification. The comparisons can be kmer-based (e.g. Kraken [25] and One Codex [15]) or alignment-based (MEGAN [6], MALT [5] and Sigma [1]). Binning methods based on kmers are usually fast, whilst alignment-based methods have greater sensitivity to distinguish the best match across similar database sequences. Benefiting from much larger databases in comparison to genomic markers used by profiling methods, binning methods usually detect more microbial species at very low abundance. However, they also tend to accumulate inaccurate assignments (false positives) [23] due to the incompleteness of the databases, resulting in reads from unrepresented taxa being erroneously attributed to multiple relatives.

While microbial species of low abundance are hard to identify by marker-based taxonomic profilers, the estimations of taxonomic bidders can be hard to interpret due to their low precision. This problem especially limits their application to the *in silico* screening of microbial content in sequenced archaeological materials [8]. Given that the ancient DNA fragments are expected to exist in low proportions in these samples, methods need to identify weak endogenous signatures hidden within a complex background that is governed by modern (environmental) contamination. Furthermore, reads from archaeological samples are fragmented and have many nucleotide mis-incorporations due to postmortem DNA damage.

We identify two challenges that limit the performance of species-level assignments. First and foremost, the reference database used for all taxonomic binnings are not comprehensive. The vast majority of microbial genetic diversity reflect uncultured organisms, which have only rarely been sequenced and analyzed. Even for the bacteria that have genomic sequences, their data are biased towards pathogens over environmental species. This leads to the next challenge where, due to the lack of proper references, reads from unknown sources can accidentally map onto distantly related references, mainly in two scenarios: (1) Foreign

reads originating from a mobile element can non-specifically map to an identical or similar mobile element in a known reference. (2) Reads originated from Ultra-Conserved Elements (UCEs), which preserve their nucleotide sequences between species, can also non-specifically map to the same UCE in an existing genome.

Addressing both of these challenges, we designed SPARSE (**S**train **P**rediction and **A**nalysis using **R**epresentative **S**Equences). In SPARSE, we index all genomes in large reference databases such as RefSeq into hierarchical clusters based on different sequence identity thresholds. A representative database that chooses one sequence for each cluster is then compiled to facilitate a fast but sensitive analysis of metagenomic samples with modest computational resources. Details are given in Sect. 2. Further, SPARSE implements a probabilistic model for sampling reads from a metagenomic sample, which extends the model described in Sigma [1] by weighting each read with its probability to stem from a genome not included in the reference database, hence considered as an unknown source. Details are given in Sect. 3.

We evaluate SPARSE on three simulated datasets published previously [14, 21, 23]. Comparing SPARSE to several other taxonomic binning software in these simulations shows its improved precision and sensitivity for assignments on the species-level or even strain-level. We further evaluate SPARSE on three ancient metagenomic datasets, demonstrating the application of SPARSE for ancient pathogen screening. For all three datasets, SPARSE is able to correctly identify small amounts of ancient pathogens in the metagenomic samples that have subsequently been confirmed by additional sequencing in the respective studies.

2 Database Indexing

2.1 Background

Average Nucleotide Identity. To catalog strain-level genomic variations within an evolutionary context, we need to reconcile all the references in a database into comprehensive classifications. Since its first publication, the average nucleotide identity (ANI) in the conserved regions of genomes has been widely used for such a purpose [10]. In particular, 95–96% ANI roughly corresponds to a 70% DNA-DNA hybridization value, which has been used for ~50 years as the definition for prokaryotic species.

Marakeby et al. [13] proposed a hierarchical clustering of individual genomes based on multiple levels of ANIs. Extending from the 95% ANI species cut-off, it allows the classification of further taxonomic levels from superkingdoms to clones. Applying such a clustering to large databases of reference genomes allows to identify clusters of overrepresented species and hence to reduce redundancy but not diversity in the database, depending the ANI levels chosen. However, the standard ANI computation adopts BLASTn [2] to align conserved regions between genomes, which is intractable to catalog large databases of reference genomes. We therefore rely on an approximation of the ANI by MASH [18] to speed-up comparisons.

ANI Approximation. MASH uses the MinHash dimensionality-reduction technique to reduce large genomes into compressed sketches. A sketch is based on a hash function applied to a kmer representation of a genome, and compression is achieved by only including the s smallest hash values of all kmers in the genome in the sketch. Comparing the sketches of two genomes, MASH defines a distance measure under a simple Poisson process of random site mutation that approximates ANI values as shown in [18].

Parameter Estimation. Ondov et al. [18] already used MASH to group all genomes in RefSeq into ANI 95% clusters. We adopted slightly different parameters and extended it to an incremental, hierarchical clustering system. The accuracy of the MASH distance approximation is determined by both the kmer length k and the sketch size s . Increasing k can reduce the random collisions in the comparison but also increase the uncertainty of the approximation. We can determine k according to equation (2) in [18]:

$$k = \lceil \log_{|\Sigma|}(n(1-q)/q) \rceil,$$

where Σ is the set of all four possible nucleotides $\{A, C, G, T\}$, n is the total number of nucleotides and q is the allowed probability of a random kmer to be found in a dataset. Given $n = 1$ terabase-pairs (Tbp; current size of RefSeq) and $q = 0.05$, which allows a 5% chance for a random k-mer to be present in a 1 Tbp database, we obtain a desired kmer size $k = 23$. Increasing the sketch size s will improve the accuracy of the approximation, but will also increase the run time linearly. We chose $s = 4000$ such that for 99.9% of comparisons that have a MASH distance of 0.05, the actual ANI values fall between 94.5–95.5%.

2.2 SPARSE Reference Database

We combine the hierarchical clustering of several ANI levels with the MASH distance computation to generate a representation of the current RefSeq [17] database. The construction of the SPARSE reference database is parallelized and incremental, thus the database can be easily updated with new genomes without a complete reconstruction.

Hierarchical Clustering. In order to cluster genomes in different levels, we defined 8 different ANI levels $L = [0.9, 0.95, 0.98, 0.99, 0.995, 0.998, 0.999, 0.9995]$ as proposed in [13], in which the genetic distances of two sequential levels differ by ~ 2 fold. The first four ANI levels differentiate strains of different species, or major populations within a species. The latter four levels give fine-grained resolutions for intra-species genetic diversities, which can be used to construct clade-specific databases for specific bacteria. In Sect. 4, we show that the first four clustering levels are sufficient for taxonomic binning down to the strain level.

The SPARSE database $D(S, L, K)$ is extended incrementally as shown in Algorithm 1, with S listing the sketches of all genomes already in the database and K being a hash containing the cluster assignments at each level $l \in L$ for

each key $s \in S$. A new genome is integrated by finding another genome in the database with the lowest distance using MASH, and clustering it with its nearest neighbour s_n depending on the ANI.

Algorithm 1. Incremental SPARSE database clustering

Input: SPARSE database $D(S, L, K)$, list of new genomes G

Output: Extended SPARSE database $D'(S, L, K)$

```

1: for each genome  $g \in G$  do
2:    $s_g = MashSketch(g)$ 
3:    $s_n = argmin_{s \in S} MashDistance(s_g, s)$ 
4:   for  $0 \leq i \leq |L| - 1$  do
5:     if  $L[i] \leq 1 - MashDistance(s_g, s_n)$  then
6:       Push  $K[s_n][i]$  to  $K[s_g]$ 
7:     else
8:       Push  $|S|$  to  $K[s_g]$ 
9:   Push  $s_g$  to  $S$ 

```

In the SPARSE implementation, we parallelized the database construction by inserting batches of genomes at once and parallelizing sketch and distance computation, thereby scaling to the complexity of the problem. After being added to the database, the cluster assignment for a genome is fixed and never redefined. Therefore, the insertion order of genomes can influence the database structure. Here we utilize prior knowledge from the community, so the SPARSE database is initialized first with all *gold standard* complete genomes in RefSeq, followed by representative and curated genomes.

Representative Database. To avoid mapping metagenomic reads to redundant genomes within the database, we construct a subset of genome representatives for read assignment, similar to [9]. The representative database consists of the first genome from each cluster defined by ANI 99%. This representative database is sufficient for routine taxonomic profiling and pathogen identification. A representative database with lower ANI values (i.e., 98% or 95%) does not recover the genetic diversities of many bacterial species and thus reduces the performance of the read-sampling model (described below). On the other hand, adding more genomes that represent finer ANI levels increases the size of the database and introduces an over-representation of references to certain pathogens. Representative databases based off these thresholds have been provided for users performing bespoke analysis of a specific species.

The representative database is then indexed using bowtie2-build [11] with standard parameters. SPARSE indexes 20,850 bacterial representative genomes in ~ 4 h using 20 computer processes. Representative databases of other ANI levels or clade-specific databases can also be built by altering the parameters. Furthermore, traditional read mapping tools such as bowtie2 [11] show reduced sensitivity for divergent reads. This is not a problem for many bacterial species, especially bacterial pathogens, because these organisms have been selectively

sequenced. However, fewer reference genomes are available for environmental bacteria and eukarya. In order to map reads from such sources to their distantly related references, SPARSE also provides an option to use MALT [5], which is slower than bowtie2 and needs extensive computing memory, but can efficiently align reads onto references with <90% similarity.

3 Metagenomic Read Sampling

Given read mappings to the representative databases as input, we adapt a probabilistic model reconstructing the process of sampling reads from a metagenomic sample to assign reads onto reference genomes. We extend the model implemented in Sigma [1] by also considering that reads aligned to a genome in the reference database could still be originating from an unknown source, thus avoiding to overestimate the number of genomes present in the sample. We introduce a weighting for each read reflecting the probability to be sampled from an unknown genome, and show in Sect. 4 how this improves the precision of taxonomic assignments.

Let E denote the set of both known and potentially unknown genomes in a metagenomic sample, and the set of reference genomes included in the SPARSE database is a subset $G \in E$. Let $Pr(r_i|E)$ be the probability of sampling a random read r_i from any possible source, we have

$$Pr(r_i | E) = Pr(r_i, G | E)Pr(r_i | G).$$

We denote $w_i = Pr(r_i, G|E)$ as the *sampling probability*, indicating the probability that r_i is sampled from any known reference genome in G . On the other hand, $Pr(r_i | G)$ is the probability of generating r_i given G and can be further separated as

$$Pr(r_i | G) = \sum_{g_j \in G} Pr(r_i | g_j)Pr(g_j | G),$$

where $Pr(g_j|G)$ is the probability that a genome $g_j \in G$ was chosen to generate the read, and $Pr(r_i|g_j)$ is the probability of obtaining read r_i from g_j . As in Sigma, given a uniform mismatch probability $\sigma = 0.05$, $Pr(r_i|g_j)$ can be directly calculated from the alignment of r_i to genome g_j with x mismatches, and can be stored in a matrix Q , such that

$$Q_{i,j} = Pr(r_i | g_j) = \sigma^x(1 - \sigma)^{l-x},$$

where l is the length of read r_i . We next describe how the sampling probability w_i is inferred, by giving a weight to each read that indicates the probability of being sampled from a known reference genome. Reads with a low weight do not influence the optimization process used to infer the optimal $Pr(g_j|G)$ for a complete metagenomic read dataset.

3.1 SPARSE Sampling Probability

We model two scenarios that can lead to non-specific mappings of foreign reads.

- (1) Since there is no systematic way of masking all mobile elements in a reference sequence, we evaluate the probability of a read being drawn from the core genome. We assume that highly conserved regions are part of the core genome, which has been vertically inherited, whereas variable regions likely represent horizontal gene transfers (HGTs). We denote this *HGT probability* as m_i .
- (2) We evaluate the probability of a read originating from an Ultra-Conserved Element (UCE), by comparing the read depths of the aligned genome fragments with other regions in the genome. UCes are so highly conserved that additional reads from divergent genomes are likely to map on to them, which results in a higher read depth than other regions. We denote this *UCE probability* as n_i . Combining both cases as a joint probability, we infer a weight w_i for each read as

$$w_i = m_i n_i.$$

HGT Probability. Given any cluster t in ANI level k that consists of u references, a read r_i can be assigned to either the core genome g_c or accessory genome g_a of this cluster. Given the number of references $v \subseteq u$ the read aligns to, we can formulate the probability of the read originating from the core genome as

$$Pr_t(g_c|r_i) = \frac{Pr_t(r_i|g_c)Pr(g_c)}{Pr(r_i)} = \frac{Pr_t(r_i|g_c)Pr(g_c)}{Pr_t(r_i|g_c)Pr(g_c) + Pr_t(r_i|g_a)(1 - Pr(g_c))},$$

$$Pr_t(r_i|g_c) = p_c^v(1-p_c)^{u-v}, \quad Pr_t(r_i|g_a) = p_a^v(1-p_a)^{u-v} \quad (1)$$

where $Pr(g_c)$ is the prior probability of any read originating from a core genomic region, and p_c and p_a are the respective probabilities for core genomic fragments or accessory genomic fragments. Default prior probabilities in SPARSE are given in Table 1. Furthermore, a read can align to multiple clusters in the same ANI level k , so we average the probabilities of all such clusters for each read weighted by Q inferred from the read alignment:

$$Pr_k(g_c|r_i) = \frac{\sum_t \max_{g_j \in t} Q_{i,j} Pr_t(g_c|r_i)}{\sum_t \max_{g_j \in t} Q_{i,j}}.$$

Finally, we consider three different ANI levels for the core genome analysis (by default 90%, 95% and 98%), assigning a lower value for m_i if the read does not map to the core genome at any of these ANI levels:

$$m_i = 1 - \prod_k (1 - Pr_k(g_c|r_i)). \quad (2)$$

Default values for the prior probabilities were inferred from a published study of core genes across multiple bacterial species [3]. We account for 1% of random deletions of core genes, which gives $p_c = 0.99$. We also observed that <10% of all genes are core genes in bacterial species represented by many genomes. This results in $\sum Pr(g_c) < 0.1$ over all three ANI levels. We arbitrarily assigned a higher $Pr(g_c)$ for levels with lower ANI, because a sequence fragment is less likely to be part of a mobile element if it is coincidentally present in more divergent genomes. Finally, ~40% of the genes in a random genome are core genes. This gives $m_i \approx 0.6$ when $v = 1$ and $u = 1$, which can be used to find empirical values of p_a via Eqs. 1 and 2.

Table 1. Default prior probabilities for three ANI levels, values inferred from [3].

ANI	$Pr(g_c)$	p_c	p_a
90%	0.05	0.99	0.1
95%	0.02	0.99	0.2
98%	0.01	0.99	0.5

UCE Probability. In order to compare the read coverage of each fragment in a reference genome g_j with other fragments of the same genome, we split its sequence into k consecutive fragments $f_{j,k}$ using two uniform arbitrary lengths, 487 bps and 2000 bps. Here 487 is used because it is a prime, such that the ends of two fragments overlap only once per Mbp. Then the read depth in each fragment, d_k , follows a Poisson distribution with parameter λ as the average number of reads per region and probability mass function $f(k, \lambda)$. Because of the complexity of the read alignments, we relax the probability of read depth in each fragment such that a wide range of read depths retain high probabilities:

$$Pr(r_i|f_{j,k}) = \begin{cases} \frac{f(d_k, \lambda/\sqrt{2})}{f(\lambda/\sqrt{2}, \lambda/\sqrt{2})} & \text{for } d_k < \lambda/\sqrt{2}, \\ 1 & \text{for } \lambda/\sqrt{2} \leq d_k \leq \sqrt{2}\lambda, \\ \frac{f(d_k, \sqrt{2}\lambda)}{f(\sqrt{2}\lambda, \sqrt{2}\lambda)} & \text{for } \sqrt{2}\lambda < d_k, \end{cases}$$

Since a read can again align to multiple genomes g_j , we compute the UCE probability of a read as a weighted average of all its alignments. If a read aligns multiple times to the same genome g_j with equal alignment score, we choose one fragment randomly. The UCE probability is then defined as

$$n_i = \frac{\sum_j (Q_{i,j} Pr(r_i|f_{j,k}))}{\sum_j Q_{i,j}}.$$

Thus a lower value of n_i is the result from a deviation of the general coverage at the read position in comparison to the average coverage in the genome, indicating that the read is likely mapping to an ultra-conserved region in the genome.

3.2 Optimization Problem

Knowing the weight w_i for all reads r_i in a whole metagenomic read set R , the task is then simplified to finding optimal $Pr(g_j|G)$ values that maximize the probability of the whole read set:

$$\max Pr(R|E) = \max \prod_{r_i \in R} Pr(r_i | E) = \max \prod_{r_i \in R} (w_i \sum_{g_j \in G} Q_{i,j} Pr(g_j|G)).$$

The optimization problem can be solved by a non-linear programming (NLP) method. In SPARSE, we rely on a modified version of the function provided in Sigma [1].

After optimizing $Pr(g_j|G)$, we finally assign a read to a potential reference by checking the following ratio of the computed probabilities:

$$P(r_i, g_j) = \frac{Pr(r_i, g_j|G)}{Pr(r_i, G)} = \frac{Q_{i,j} * Pr(g_j|G)}{\sum_{g_j \in G} Q_{i,j} * Pr(g_j|G)}. \quad (3)$$

We may assign a read to multiple references, as long as $\frac{P(r_i, g_j)}{\max_g P(r_i, g)} \geq 0.1$. This allows a better abundance estimation for multiple strains from the same species, in which case a read cannot be assigned unambiguously to a single reference.

Further, let $r_i \in B \subset R$ be all reads assigned to g_j . For a read r_i of length l with x mismatches in the alignment to its assigned reference, we have a nucleotide similarity of $s_{i,j} = \frac{l-x}{l}$. The weighted average similarity $\bar{s}_{B,j}$ can be calculated as

$$\bar{s}_{B,j} = \frac{\sum_{r_i \in B} s_{i,j} w_i P(r_i, g_j)}{\sum_{r_i \in B} w_i P(r_i, g_j)}.$$

Potentially, reads assigned to a single reference could still originate from several co-existing genomes, with varying degrees of diversity, in the metagenome. We can identify reads from more divergent sources by comparing $s_{i,j}$ to their average similarity. If all reads assigned to a single reference originate from the same genome in the metagenome, we assume that the similarity of most reads complies with the average similarity over all reads. However, reads originating from very conserved regions show higher similarity than the average and provide a sampling bias. On the other hand, reads originating from different more divergent genomes, will show lower similarity which can be used to avoid over-estimating the abundance of each cluster. Therefore we compute the expected average nucleotide identity s' for r_i as

$$s'_{i,j} = \min(s_{i,j}, \bar{s}_{B,j}).$$

This similarity reflects the ANI between each read and the assigned reference and, as described in the next section, can be used to compute the abundance of each cluster in the metagenomic sample.

3.3 ANI Cluster Abundances

The equation $m_i n_i P(r_i, g_j)$ describes the probability, for each read $r_i \in R$, to be drawn from a region in reference g_j that is part of the core genome (m_i) and has even read depth in comparison to the whole chromosome (n_i). In summary for all reads assigned to g_j , $\sum_i m_i n_i * P(r_i, g_j)$ gives the frequency of reads originating from the core genome of g_j . However, the desired read abundance for a reference g_j needs to also include reads from the accessory genome. Such reads have been previously suppressed when computing m_i . If we assume that all species have the same proportion of core genome, the relative abundances of their core genomes will be equal to the relative abundance of their whole genomes. However, since this is not the case [3], we need to normalize each m_i computed previously. Given $P(r_i, g_j)$ from Eq. 3, for any ANI 90% cluster t , we normalize m_i for a read r_i as

$$m'_i = \frac{\sum_{g_j \in t} \sum_{\substack{r_k \in R, \\ s'_{k,j} \geq 0.9}} P(r_k, g_j)}{\sum_{g_j \in t} \sum_{\substack{r_k \in R, \\ s'_{k,j} \geq 0.9}} m_k P(r_k, g_j)} * m_i.$$

Finally, we assign reads into clusters of all ANI levels according to the references contained in the cluster. For each cluster, we only assign reads if its similarity complies with the ANI level l of the cluster, i. e. $s'_{i,j} \geq l$.

Thus the abundance of a cluster t_l is computed as the sum of all read abundances assigned to all genomes in the cluster weighted by their probability to originate from an unknown genome. Therefore clusters containing only reads with small n_i and m_i probabilities will receive a low abundance value even if many reads are assigned to it.

$$a_{t_l} = \sum_{g_j \in t_l} \sum_{\substack{r_i \in R \\ s'_{i,j} \geq l}} m'_i n_i P(r_i, g_j).$$

3.4 Taxonomic Labels for ANI Clusters

We finally assign standard taxonomic designations to all clusters at all ANI levels, in order to interpret their biological meaning. Here we rely on a majority vote of all genomes in a cluster. However, the taxonomic levels are restricted to certain ANI levels. For example, species are distinguished at the ANI 95% level, and a species designation is therefore inappropriate for an ANI 90% cluster. Similarly, the taxonomic label for an ANI 95% cluster should not include any subspecies designations.

4 Evaluation

4.1 Representative Database

We ran SPARSE to index the RefSeq database that consists of 101,680 complete or draft genomes into 28,732 clusters at ANI 99% level, which were further

grouped into 18,205 clusters at 95% ANI level, as shown in Fig. 1. Grouping all the genomes according to their species, the resulting representative database is much more evenly distributed, with a Pielou’s evenness [19] of $J' = 0.9$, comparing to $J' = 0.51$ for the whole RefSeq database. Over-representation of pathogenic organisms in the RefSeq database are largely due to repeated sequencing of nearly identical genomes rather than sequencing of intra-species genetic diversities. In particular, nearly half of the genomes in RefSeq are from the top 10 most sequenced bacterial species, which are all human pathogens. All these genomes were grouped into 615 clusters at ANI 99% level, which gives a 65-fold reduction of the data indexed for these species. With this strategy, the whole RefSeq database was downloaded and assigned into ANI levels in ~ 23 h, using 20 processes on a standalone server. Further insertion of 1,000 new genomes (~ 5 MB) into an already established database takes ~ 15 mins.

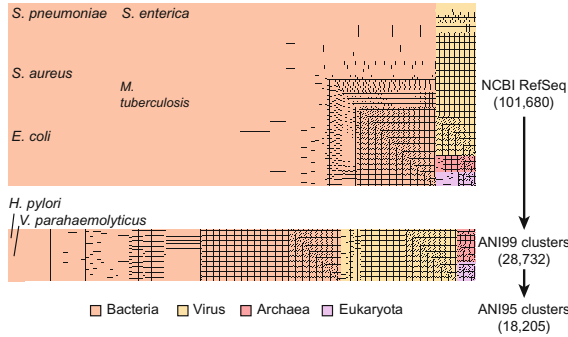


Fig. 1. Hierarchical clustering of 101,680 genomes in NCBI RefSeq database (Aug. 2017) into 18,205 ANI 95% clusters using SPARSE. Each rectangle represents such a cluster at ANI 95% level, with its area relative to the total number of genomes (top) or clusters at ANI 99% (bottom).

4.2 Simulated Data

We ran SPARSE on three recent simulated datasets (Sczyrba et al. [23], McIntyre et al. [14] and Quince et al. [21]). For a fair comparison, the analyses for all datasets were based on a database built from NCBI RefSeq and taxonomy databases dated 22th June, 2015, which is the deadline for the comparison in [23] and also pre-dates the other two comparisons. We evaluated the performance of SPARSE as described in the respective papers for the read-level taxonomic binners, adopting the results for the other compared methods directly from the studies. We additionally included Sigma using the same database as SPARSE in the comparison. We calculated sensitivity and precision based on the number of true-positives (TP; correctly assigned reads), false-positives (FP; incorrectly assigned reads), and false-negatives (FN; unassigned reads).

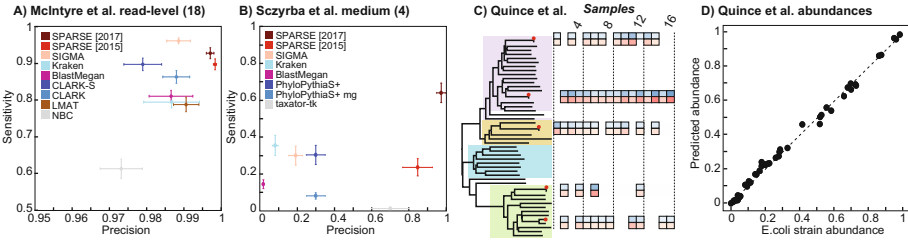


Fig. 2. Performances of SPARSE in simulated published datasets. The performance of all the tools in A and B, except for SPARSE and Sigma, are obtained from the respective publications [14, 23]. SPARSE was run in parallel using two different databases. [2015] uses database built from RefSeq at 2015, whereas [2017] uses up-to-date database. (A) All the simulated reads in McIntyre et al. [14] were derived from published genomes. (B) The Sczyrba et al. [23] used unpublished genomes for read simulations. C+D) Strain-level identification using the mocked *E. coli* datasets as published in [21]. (C) Left: The distance-based species tree for *E. coli* for 45 ANI 99% representative genomes plus the five genomes used in [21] for mocked reads. The four largest ANI 98% clusters in *E. coli* are highlighted with colors. Right: Each column shows one of the 16 mocked samples. The true relative abundances of *E. coli* strains in samples (blue) and the relative abundances of predicted strains (red) in samples are shown as colored squares. (D) Comparison of true *E. coli* strain abundances versus SPARSE predictions. The dashed line indicates the linear regression of the two values, with $R^2 = 0.9948$ and $p < 2.2e-16$. (Color figure online)

All simulated reads in the McIntyre et al. [14] study were generated from published complete genomes. This dataset is suitable for comparing the completeness of the databases, as well as the sensitivity of the read mapping approaches in different tools. Both SPARSE and Sigma were run on 18 samples that have read-level taxonomic labels. SPARSE binned all the samples in ~ 10 hours with 20 processes. The precision and sensitivity of both tools in addition to six binning tools from [14] are summarized in Fig. 2A. As expected, all tools reached a high precision of $>97\%$, but differed in their sensitivity. Benefiting from the representative database, SPARSE and Sigma assigned the highest numbers of reads into correct species. The difference between the two methods is due to their different strategies in the modeling, where Sigma assigned all reads to their possible references, whereas SPARSE filtered out unreliable mappings. An independent run of SPARSE using the latest RefSeq database (Aug. 2017) assigned slightly more reads into species, but does not improve precision. This database consists of 20,850 representative genomes, which is ~ 2 fold the number of representatives (9,707) in RefSeq 2015. The run time of SPARSE increases with this database to ~ 24 h, which is also ~ 2 times slower as running SPARSE against RefSeq 2015.

The datasets in Sczyrba et al. [23] are much more challenging, because all the reads were generated from sequencing of environmental isolates, many of which do not have closely related references in the 2015 database. Furthermore, many reads do not have a known microbial species label, because they are not similar

to any species in SILVA [20], which was used as the gold standard in this study. We ran both Sigma and SPARSE on the medium complexity datasets, and compared the results with the other methods (see Fig. 2f in [23]) for the recovery of microbial species (Fig. 2B). Using 80 processes, SPARSE ran through all four datasets in ~ 40 h. All the taxonomic bidders published in [23] obtained an average precision of $<30\%$ at species level, except for taxator-tk [4] with a precision of 70% along with the lowest sensitivity ($\sim 1.25\%$). The performance of Sigma is comparable to other binning tools, whereas SPARSE obtained an exceptionally high precision of $\sim 85\%$ while still maintaining a sensitivity of $\sim 23\%$. Many incorrect taxonomic bins predicted in Sigma were suppressed in SPARSE, because they have low sampling probability w_i to any of the existing references. Again, SPARSE was also run independently against the database built Aug. 2017. The runs completed in 4 days and recovered 63% of the species in the CAMI median datasets, with an average precision of 97% .

Both benchmarks evaluate the performances of taxonomic binnings on or above species level, but give no resolution in intra-species diversity. DESMAN [21] allows reference-free recovery of strain-level variations based on uneven read depths of different strains across multiple samples. It has been compared with two other strain-level binning methods using mock *E. coli* samples [21]. Applying SPARSE to the same 20 genome mocks, we recovered 50/51 *E. coli* strains in all 16 samples without any additional strains (false positives), as shown in Fig. 2C. The only strain that was not recovered by SPARSE is 2011C-3493 in the 12th sample (Sample733 in [21]), which accounts for only $\sim 0.03\%$ of all *E. coli* reads in the sample. We also obtained an almost exact correspondence between the relative abundances of the strains and the predictions (Fig. 2D). A linear regression of real abundances and the predictions gives an $R^2 = 0.9948$ and $p < 2.2e-16$.

4.3 Ancient Metagenomes

We further evaluated SPARSE and five additional metagenomic tools on three real sets of ancient DNA reads (*Mycobacterium tuberculosis* from [7], *Yersinia pestis* from [22] and *Helicobacter pylori* from [12]) and summarised their results in Table 2. For all samples, the presence of the targeted pathogen, although in very low frequencies ($\leq 0.02\%$), has been confirmed by additional sequencing in the respective publications. MIDAS [16] failed in all three samples and MetaPhlan2 [24] managed to identify *H. pylori* but failed in the other two samples. The results for these two marker-based approaches are consistent with the simulations discussed earlier. Kraken [25] and One Codex [15] are both based on kmer-based taxonomic assignment, but yielded different results. Kraken only identified *H. pylori*, whereas One Codex got positive results in all three samples. However both methods incurred a high number of false positives. For example, Kraken reported *Salmonella enterica* and *Vibrio cholerae* in the Iceman sample, whereas One Codex predicted two *Yersinia* species. All these predictions are inconsistent with results from other tools and analyses presented in the publications. Sigma identified two of three pathogens but inaccurately predicted

V. parahaemolyticus, which is normally associated with seafood, for the human remains from the Bronze Age. SPARSE successfully identified all three targeted species without any additional suspicious pathogen, which highlights its application to archaeological samples.

It took SPARSE ~ 1 and ~ 2.5 h to profile the *M. tuberculosis* and *Y. pestis* datasets respectively, and over 16 h for the *H. pylori* dataset, using 20 processes in a standalone server. The run-time for Sigma are approximately 5-fold higher than SPARSE in all the datasets. For both tools, the read alignment is the main limiting factor and accounted for over $\sim 95\%$ of their run-time. In contrast, the other binning tools listed in the table finished within 10 min on all the datasets, due to their different ways of handling reads.

Table 2. Summary of results for different metagenomic binning tools on real archaeological datasets identifying ancient pathogens.

	ERR650978 [7] 1794AD Hungarian 1.7M reads MT 0.02%	ERR1094783 [12] 5300-yr-old Iceman 15M reads <i>H. pylori</i> 0.01%	ERR1018927 [22] Bronze Age human 1.6M reads <i>Y. pestis</i> 0.01%
SPARSE	+	+	+
Sigma	+	–	+(VP)
Kraken	– (CD,ML)	+(SE,VC)	–
One Codex	+(MA,SA)	+(YE,YP)	+
MetaPhlan	–	+	–
MIDAS	–	–	–

^a +/– for the identification of the pathogen. Abbreviations for suspicious predictions in bracket (CD: *Corynebacterium diphtheriae*; MA: *M. avium*; ML: *M. leprae*; MT: *M. tuberculosis*; SA: *Staphylococcus aureus*; SE: *S. enterica*; VC: *V. cholerae*; VP: *V. parahaemolyticus*; YE: *Y. enterocolitica*; YP: *Y. pseudotuberculosis*).

5 Conclusion

The genetic signatures of specific microbes in metagenomic data, such as human pathogens, are often buried behind the majority of reads from genetically diverse environmental organisms. This is exemplified in the metagenomic sequencing of archaeological samples. Current taxonomic assignment methods compare the metagenomic data with databases that do not fully capture the diversity of microbial genomes. Among these tools, the marker-based taxonomic profilers fail to identify species at low abundances whereas whole genome based taxonomic bidders give inaccurate predictions due to non-specific read mappings on ultra-conserved or horizontally transferred elements.

SPARSE indexes existing reference genomes into a comprehensive database with automatic hierarchical clusterings of related organisms. This database is used as a reference for mapping of metagenomic reads. SPARSE penalizes unreliable mappings of reads from unknown sources, and integrates all remaining into a probabilistic model, in which reads are assigned to either an existing reference or unknown sources. In both simulations and real archaeological data, SPARSE outperforms all existing methods, especially in the precision of species-level assignment. Furthermore, SPARSE manages to identify multiple strains of the same species even when they co-exist in the same sample. In contrast to many existing tools, SPARSE aligns metagenomic reads onto a huge representative database. This database, albeit being a compression of the even larger RefSeq database, is still much larger than many existing databases. As a result, the run-time of SPARSE is limited by the performance of its adopted read aligner, which could be improved in the future development.

Acknowledgements. M.A., Z.Z., N.L. and N-F.A. were supported by Wellcome Trust (202792/Z/16/Z). Additional initial grant support was from BBSRC (BB/L020319/1).

References

1. Ahn, T.H., Chai, J., Pan, C.: Sigma: strain-level inference of genomes from metagenomic analysis for biosurveillance. *Bioinformatics* **31**(2), 170–177 (2015)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* **215**(3), 403–410 (1990)
3. Ding, W., Baumdicker, F., Neher, R.A.: panX: pan-genome analysis and exploration. *bioRxiv* 10.1101/072082 (2016)
4. Dröge, J., Gregor, I., McHardy, A.C.: Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics* **31**(6), 817–824 (2014)
5. Herbig, A., Maixner, F., Bos, K.I., Zink, A., Krause, J., Huson, D.H.: Malt: fast alignment and analysis of metagenomic DNA sequence data applied to the Tyrolean Iceman. *bioRxiv* 10.1101/050559 (2016)
6. Huson, D.H., Beier, S., Flade, I., Górska, A., El-Hadidi, M., Mitra, S., Ruscheweyh, H.J., Tappu, R.: MEGAN community edition-interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Comput. Biol.* **12**(6), e1004957 (2016)
7. Kay, G.L., Sergeant, M.J., Zhou, Z., Chan, J.Z.M., Millard, A., Quick, J., Szikossy, I., Pap, I., Spigelman, M., Loman, N.J., Achtman, M., Donoghue, H.D., Pallen, M.J.: Eighteenth-century genomes show that mixed infections were common at time of peak tuberculosis in Europe. *Nat. Commun.* **6**, 6717 (2015)
8. Key, F.M., Posth, C., Krause, J., Herbig, A., Bos, K.I.: Mining metagenomic data sets for ancient DNA: recommended protocols for authentication. *Trends Genet.* **33**(8), 508–520 (2017)
9. Kim, D., Song, L., Breitwieser, F.P., Salzberg, S.L.: Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**(12), 1721–1729 (2016)
10. Konstantinidis, K.T., Tiedje, J.M.: Genomic insights that advance the species definition for prokaryotes. *Proc. Natl. Acad. Sci.* **102**(7), 2567–2572 (2005)

11. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**(4), 357–359 (2012)
12. Maixner, F., Krause-Kyora, B., Turaev, D., Herbig, A., Hoopmann, M.R., Hallows, J.L., Kusebauch, U., Vigel, E.E., Malferttheiner, P., Megraud, F., et al.: The 5300-year-old *Helicobacter pylori* genome of the Iceman. *Science* **351**(6269), 162–165 (2016)
13. Marakeby, H., Badr, E., Torkey, H., Song, Y., Leman, S., Monteil, C.L., Heath, L.S., Vinatzer, B.A.: A system to automatically classify and name any individual genome-sequenced organism independently of current biological classification and nomenclature. *PLoS One* **9**(2), e89142 (2014)
14. McIntyre, A.B.R., Ounit, R., Afshinnekoo, E., Prill, R.J., Hénaff, E., Alexander, N., Minot, S.S., Danko, D., Fook, J., Ahsanuddin, S., et al.: Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol.* **18**(1), 182 (2017)
15. Minot, S.S., Krumm, N., Greenfield, N.B.: One Codex: A Sensitive and Accurate Data Platform for Genomic Microbial Identification. *bioRxiv* 10.1101/027607 (2015)
16. Nayfach, S., Rodriguez-Mueller, B., Garud, N., Pollard, K.S.: An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography. *Genome Res.* **26**(11), 1612–1625 (2016)
17. O’Leary, N.A., Wright, M.W., Brister, J.R., Ciufu, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., et al.: Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**(D1), D733–D745 (2015)
18. Ondov, B.D., Treangen, T.J., Melsted, P., Mallonee, A.B., Bergman, N.H., Koren, S., Phillippy, A.M.: Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.* **17**(1), 132 (2016)
19. Pielou, E.C.: *Ecological Diversity*. Wiley, New York (1975)
20. Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., Glöckner, F.O.: The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res.* **41**(D1), D590–D596 (2012)
21. Quince, C., Delmont, T.O., Raguideau, S., Alneberg, J., Darling, A.E., Collins, G., Eren, A.M.: DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biol.* **18**(1), 181 (2017)
22. Rasmussen, S., Allentoft, M.E., Nielsen, K., Orlando, L., Sikora, M., Sjögren, K.G., Pedersen, A.G., Schubert, M., Van Dam, A., Kapel, C.M.O., et al.: Early divergent strains of *Yersinia pestis* in Eurasia 5,000 years ago. *Cell* **163**(3), 571–582 (2015)
23. Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., Gregor, I., Majda, S., Fiedler, J., Dahms, E., Bremges, A., Fritz, A., Garrido-Oter, R., Jørgensen, T.S., et al.: Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nat. Methods* **14**(11), 1063 (2017)
24. Truong, D.T., Franzosa, E.A., Tickle, T.L., Scholz, M., Weingart, G., Pasolli, E., Tett, A., Huttenhower, C., Segata, N.: Metaphlan2 for enhanced metagenomic taxonomic profiling. *Nat. Methods* **12**(10), 902–903 (2015)
25. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.* **15**(3), R46 (2014)
26. Zhou, Z., Lundstrøm, I., Tran-Dien, A., Duchêne, S., Alikhan, N.F., Sergeant, M.J., Langridge, G., Fotakis, A.K., Nair, S., Stenøien, H.K., et al.: Millennia of genomic stability within the invasive Para C Lineage of *Salmonella enterica*. *bioRxiv* 10.1101/105759 (2017)

Short Papers



Targeted Genotyping of Variable Number Tandem Repeats with adVNTR

Mehrdad Bakhtiari¹(✉), Sharonna Shleizer-Burko², Melissa Gymrek^{1,2},
Vikas Bansal³, and Vineet Bafna¹

¹ Department of Computer Science and Engineering,
University of California, San Diego, La Jolla, CA 92093, USA
mbakhtiari@ucsd.edu, vbafna@eng.ucsd.edu

² Department of Medicine, University of California,
San Diego, La Jolla, CA 92093, USA

³ Department of Pediatrics, University of California,
San Diego, La Jolla, CA 92093, USA

Extended Abstract

Whole Genome Sequencing is increasingly used to identify Mendelian variants in clinical pipelines. These pipelines focus on single nucleotide variants (SNVs) and also structural variants, while ignoring more complex repeat sequence variants. We consider the problem of genotyping *Variable Number Tandem Repeats* (VNTRs), composed of inexact tandem duplications of short (6–100 bp) repeating units. VNTRs span 3% of the human genome, are frequently present in coding regions, and have been implicated in multiple Mendelian disorders (*e.g.*, Medullary cystic kidney disease, Myoclonus epilepsy, and FSHD) and complex disorders such as bipolar disorder. In some cases, the disease associated variants correspond to point mutations in the VNTR sequence while in other cases, changes in the number of tandem repeats (RU count) show a statistical association (or causal relationship) with disease risk. While existing tools are able to recognize VNTR carrying sequence, genotyping VNTRs (determining repeat unit count and sequence variation) from whole genome sequenced reads remains challenging. We describe a method, adVNTR, that models the problems of RU counting and mutation detection using HMMs trained for each target VNTR. adVNTR models can be developed for short-read (Illumina) and single molecule (PacBio) whole genome and exome sequencing. It has three components: (i) HMM training module for model parameter estimation; (ii) read recruitment; and, (iii) estimating RU counts and variant detection. We compared read recruitment with alignment-based methods. The results show that while adVNTR works well for a range of RU counts, other mapping tools work well only when the simulated RU count matches the reference RU count. We performed a long range (LR)PCR experiment on the individual NA12878 to assess the accuracy of the adVNTR genotypes. To test performance of counting of Repeat Units on real data where the true VNTR genotype is not known, we confirmed our results by checking for Mendelian inheritance consistency at 865 VNTRs in two trios.

For short VNTRs, adVNTR can be an effective tool for larger population-scale studies of VNTR genotypes using WGS data replacing labor intensive gel electrophoresis. We found the RU count frequencies for two disease-linked VNTRs in GP1BA and MAOA genes, using 150 PCR-free WGS data. The 2R/3R genotypes in GP1BA are associated with Aspirin Treatment failure for stroke prevention. Notably, our results suggest that the 2R genotype is absent in African populations suggesting that this shorter allele arose after the out of Africa transition. adVNTR is available at <https://github.com/mehrdadbakhtiari/adVNTR>.

Reference

1. Bakhtiari, M., Shleizer-Burko, S., Gymrek, M., Bansal, V., Bafna, V.: Targeted genotyping of variable number tandem repeats with adVNTR. bioRxiv, p. 221754 (2017)



Positive-Unlabeled Convolutional Neural Networks for Particle Picking in Cryo-electron Micrographs

Tristan Bepler^{1,2}, Andrew Morin^{2,6}, Alex J. Noble³, Julia Brasch⁴,
Lawrence Shapiro^{4,5}, and Bonnie Berger^{1,2,6}(✉)

¹ Computational and Systems Biology, MIT, Cambridge, MA, USA
bab@mit.edu

² Computer Science and AI Laboratory, MIT, Cambridge, MA, USA

³ National Resource for Automated Molecular Microscopy, Simons Electron Microscopy Center, New York Structural Biology Center, New York, NY, USA

⁴ Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY, USA

⁵ Mortimer B. Zuckerman Mind Brain Behavior Institute, New York, NY, USA

⁶ Department of Mathematics, MIT, Cambridge, MA, USA

Background

Structure determination with cryoEM involves reconstructing a 3D molecule from 2D projections. This process often requires tens to hundreds of thousands of experimental projections, or particles. Locating these particles in cryoEM micrographs, referred to as particle picking, is a major bottleneck in the current protein structure determination pipeline. This pipeline generally consists of sample and EM grid preparation, imaging, particle picking, and eventually structure determination. Labeling a sufficient number of particles to determine a high resolution structure can require months of effort – even with the use of existing methods designed to automate the process. Limitations of these tools include high false positive rates, requiring many hand-labeled training examples, and poor performance on non-globular proteins.

In order to better automate particle picking, and thus accelerate structure determination, we newly frame the particle picking problem as an instance of positive-unlabeled classification. In our framework, for a set of micrographs containing particles of interest with a small number labeled for training, we learn a convolutional neural network (CNN) to classify particles from background using a novel generalized-expectation criteria [1] to regularize the model's posterior over the unlabeled micrograph regions. This advance allows us to achieve state-of-the-art particle detection results with minimal hand-labeling required.

B.Berger — This work was partially supported by grants: NIH R01-GM081871, NIH R01-MH1148175, Simons Foundation (349247), NYSTAR, NIH NIGMS (GM103310), the Agouron Institute (F00316) and NIH S10 OD019994-01.

Methods

We develop Topaz, the first particle picking pipeline to use CNNs trained using only positive and unlabeled examples and GE-binomial, a general objective function for learning classifier parameters from positive and unlabeled data. The GE-binomial objective penalizes the negative log-likelihood of the labeled data points while regularizing the classifier’s posterior over the unlabeled data to match a binomial distribution prior on the number of unlabeled positives. Denoting the set of labeled positive data points by P , the probabilistic classifier as g , the classifier’s posterior over the number of unlabeled positives as q , and the binomial prior as p , the GE-binomial objective function is: $-\mathbb{E}_{x \in P} [\log g(x)] + KL(q \parallel p)$,

where KL is the Kullback-Leibler divergence.

In the Topaz pipeline, CNN classifiers are fit to labeled particles and the remaining unlabeled micrograph regions using minibatched stochastic gradient descent to minimize the GE-binomial objective. Predicted particle coordinates are next extracted by scoring each micrograph region with the trained classifier and then using the non-maximum suppression algorithm to greedily select candidate particle coordinates.

Results

We show that the Topaz pipeline is able to accurately detect particles when trained with very few labeled example particles. On the EMPIAR-10096 cryoEM data set [2], Topaz achieves 46% precision at 90% recall with only 1000 labeled particles. In contrast, at the same recall level, EMAN2’s byRef method [3] only reaches 33% precision with the same set of labeled particles – corresponding to 71% more false positives than Topaz. Remarkably, Topaz still achieves better precision than EMAN2 at 90% recall with 1/10th and even 1/100th the number of labeled particles. At all numbers of labeled particles tested, we improve substantially over EMAN2’s byRef method in area under the precision-recall curve. The relative improvement in particle detection provided by Topaz is even greater on a second, unpublished dataset provided by the Shapiro lab, containing stick-like particles with low signal-to-noise ratio. Furthermore, we show that combining a convolutional decoder with the convolutional feature extractor and classifier learned with GE-binomial to form a hybrid classifier+autoencoder can further improve generalization when very few labeled data points are available. Finally, we demonstrate that our GE-binomial objective function outperforms other positive-unlabeled learning methods never before applied to particle picking. Topaz runs efficiently, training in hours and predicting in seconds with a single consumer grade GPU. We expect Topaz to become an essential component of single particle cryoEM analysis and our GE-binomial objective function to be widely applicable to positive-unlabeled classification problems.

References

1. Mann, G.S., McCallum, A.: Generalized expectation criteria for semi-supervised learning with weakly labeled data. *J. Mach. Learn. Res.* **11**, 955–984 (2010)
2. Tan, Y.Z., Baldwin, P.R., Davis, J.H., Williamson, J.R., Potter, C.S., Carragher, B., Lyumkis, D.: Addressing preferred specimen orientation in single-particle cryo-EM through tilting. *Nat. Methods* **14**, 793–796 (2017). <https://doi.org/10.1038/nmeth.4347>
3. Tang, G., Peng, L., Baldwin, P.R., Mann, D.S., Jiang, W., Rees, I., Ludtke, S.J.: EMAN2: an extensible image processing suite for electron microscopy. *J. Struct. Biol.* **166**, 205–213 (2007). <https://doi.org/10.1016/j.jsb.2006.05.009>



Designing RNA Secondary Structures Is Hard

Édouard Bonnet¹, Paweł Rzażewski², and Florian Sikora³(✉)

¹ Department of Computer Science, Middlesex University, London, UK
`edouard.bonnet@dauphine.fr`

² Faculty of Mathematics and Information Science,
Warsaw University of Technology, Warsaw, Poland
`p.rzazewski@mini.pw.edu.pl`

³ Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE,
Paris, France
`florian.sikora@dauphine.fr`

An RNA sequence is a word over an alphabet on four elements $\{A, C, G, U\}$ called bases. RNA sequences fold into secondary structures where some bases pair with one another while others remain unpaired. Pseudoknot-free secondary structures can be represented as well-parenthesized expressions with additional dots, where pairs of matching parentheses symbolize paired bases and dots, unpaired bases. The two fundamental problems in RNA algorithmic are to *predict* how sequences fold within some model of energy and to *design* sequences of bases which will fold into targeted secondary structures. Predicting how a given RNA sequence folds into a pseudoknot-free secondary structure is known to be solvable in cubic time since the eighties [15, 16] and in truly subcubic time by a recent result of Bringmann et al. [3], whereas Lyngsø has shown it is NP-complete if pseudoknots are allowed [13]. As a stark contrast, it is unknown whether or not designing a given RNA secondary structure is a tractable task; this has been raised as a challenging open question by several authors [2, 6, 7, 9, 11, 14]. Because of its crucial importance in a number of fields such as pharmaceutical research and biochemistry, there are dozens of heuristics and software libraries dedicated to RNA secondary structure design [1, 2, 4, 5, 8]. It is therefore rather surprising that the computational complexity of this central problem in bioinformatics has been unsettled for decades.

As our main result we show that, in the simplest model of energy which is the Watson-Crick model the design of secondary structures is NP-complete if one adds natural constraints of the form: *index i of the sequence has to be labeled by base b* . This negative result suggests that the same lower bound holds for more realistic models of energy. It is noteworthy that the additional constraints are by no means artificial: they are provided by all the RNA design pieces of software and they do correspond to the actual practice (see for example the instances of the EteRNA project [12]). Our reduction from a variant of 3-SAT has as main ingredients: arches of parentheses of different widths, a linear order interleaving variables and clauses, and an intended *rematching strategy* which increases the number of pairs if and only if the three literals of a same clause are false. The correctness of the construction is also quite intricate; it relies on the polynomial

algorithm for the design of saturated structures – secondary structures without dots – by Haleš et al. [9, 10], counting arguments, and a concise case analysis.

We also show that a naive brute-force algorithm for RNA DESIGN can be improved by a careful structural analysis.

References

1. Aguirre-Hernández, R., Hoos, H.H., Condon, A.: Computational RNA secondary structure design: empirical complexity and improved methods. *BMC Bioinf.* **8**(1), 34 (2007)
2. Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H., Condon, A.: A new algorithm for RNA secondary structure design. *J. Mol. Biol.* **336**(3), 607–624 (2004)
3. Bringmann, K., Grandoni, F., Saha, B., Williams, V.V.: Truly sub-cubic algorithms for language edit distance and RNA-folding via fast bounded-difference min-plus product. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, pp. 375–384 (2016)
4. Butterfoss, G.L., Kuhlman, B.: Computer-based design of novel protein structures. *Annu. Rev. Biophys. Biomol. Struct.* **35**, 49–65 (2006)
5. Churkin, A., Retwitzer, M. D., Reinharz, V., Ponty, Y., Waldispühl, J., Barash, D.: Design of RNAs: comparing programs for inverse RNA folding. *Briefings Bioinf.* (2017)
6. Condon, A.: Problems on RNA secondary structure prediction and design. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 22–32. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_2
7. Condon, A.: RNA molecules: glimpses through an algorithmic lens. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 8–10. Springer, Heidelberg (2006). https://doi.org/10.1007/11682462_2
8. García-Martín, J. A., Clote, P., Dotú, I.: RNAiFold: a web server for RNA inverse folding and molecular design. *Nucleic Acids Res.* **41**(Webserver-Issue), 465–470 (2013)
9. Hales, J., Héliou, A., Manuch, J., Ponty, Y., Stacho, L.: Combinatorial RNA design: designability and structure-approximating algorithm in watson-crick and nussinov-jacobson energy models. *Algorithmica* **79**(3), 835–856 (2017)
10. Haleš, J., Maňuch, J., Ponty, Y., Stacho, L.: Combinatorial RNA design: designability and structure-approximating algorithm. In: Cicalese, F., Porat, E., Vaccaro, U. (eds.) CPM 2015. LNCS, vol. 9133, pp. 231–246. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19929-0_20
11. Jedwab, J., Petrie, T., Simon, S.: An infinite class of unsaturated rooted trees corresponding to designable RNA secondary structures. *CoRR*, abs/1709.08088 (2017)
12. Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Gaikwad, S., Yoon, S., Treuille, A., Das, R., Participants, E.R.N.A.: RNA design rules from a massive open laboratory. *Proc. Nat. Acad. Sci.* **111**(6), 2122–2127 (2014)
13. Lyngsø, R.B.: Complexity of pseudoknot prediction in simple models. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 919–931. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27836-8_77

14. Lyngsø, R.B.: Inverse folding of RNA (2012). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.226.5439&rep=rep1&type=pdf>
15. Nussinov, R., Jacobson, A.B.: Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Nat. Acad. Sci.* **77**(11), 6309–6313 (1980)
16. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* **9**(1), 133–148 (1981)



Generalizable Visualization of Mega-Scale Single-Cell Data

Hyunghoon Cho¹, Bonnie Berger^{1,2}(✉), and Jian Peng³(✉)

¹ CSAIL, MIT, Cambridge, MA 02139, USA

² Department of Mathematics, MIT, Cambridge, MA 02139, USA
bab@mit.edu

³ Department of Computer Science, UIUC, Urbana, IL 61801, USA
jianpeng@illinois.edu

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has been a key tool in dissecting inter-cellular variation in biomedical sciences. A standard analysis for scRNA-seq data is to visualize the cells in a low-dimensional (2D or 3D) space via methods such as *t-stochastic neighbor embedding* (t-SNE) [1], where each cell is represented as a dot and dots of cells with similar expression profiles are located close to each other in space. Such visualization reveals the salient structure of the data in a form that is easy for researchers to grasp and further analyze.

Recent advances in sequencing technologies has led to an exponential growth in the number of cells sequenced in a study. For example, 10x Genomics recently published a dataset of 1.3 million mouse neurons [2]. The emergence of such *mega-scale* data poses new computational challenges before they can be widely adopted, as many of the existing tools for scRNA-seq analysis (including t-SNE) require prohibitive runtimes or computational resources for data of this size.

We introduce *neural t-SNE* (net-SNE), a scalable and generalizable method for visualizing millions of cells for scRNA-seq analysis. net-SNE learns a high-quality mapping *function* that takes an expression profile as input and outputs a low-dimensional embedding in 2D or 3D for visualization. Unlike t-SNE, the mapping function learned by net-SNE can be used to map *previously unseen* cells. In addition to allowing fast visualization of datasets with millions of cells, net-SNE enables novel workflows for single-cell genomics, where newly observed cells are visualized in the context of existing datasets for translational analysis.

2 Methods

Our method (net-SNE) models the position of each cell in the visualization as the output of a parameterized map evaluated at the given expression profile. We use feedforward neural networks (NNs) to represent the embedding function, drawing from the intuition that NNs have sufficient expressive capacity to find high-quality maps similar to those typically uncovered by t-SNE. To optimize

the NN parameters, net-SNE minimizes the same objective score optimized by t-SNE via gradient descent. This choice of objective allows net-SNE to emulate the behavior of t-SNE while newly achieving generalizability and scalability. Notably, net-SNE is compatible with existing optimizations for t-SNE—our implementation of net-SNE incorporates an efficient variant of t-SNE based on Barnes-Hut approximation [1]. We achieve further efficiency by employing stochastic optimization techniques, where only a subset of cells are used to approximate each parameter update. Such stochastic acceleration is newly enabled by net-SNE due to the fact that parameters being optimized are *shared* across all cells.

3 Results

We observed that net-SNE learns an embedding that closely matches t-SNE on 13 scRNA-seq datasets with known clusters in terms of both visual quality and clustering accuracy. Furthermore, when an entire cluster of cells was withheld and placed onto the visualization after the fact, net-SNE accurately positioned the held-out cells as a distinct cluster, despite not having seen any cells from the missing cluster. To demonstrate fast visualization of mega-scale datasets, we also pre-trained net-SNE on a random subset of 100K cells from the 10x Genomics dataset and used the learned embedding to instantly visualize the entire dataset in less than a minute. This approach obtained a higher quality map than t-SNE with the default parameters, the latter of which took 13h to finish. While the pre-training of net-SNE took 3h in our experiment, we note that a pre-trained embedding may be readily available in certain use cases. We provide example visualizations by net-SNE in Fig. 1.

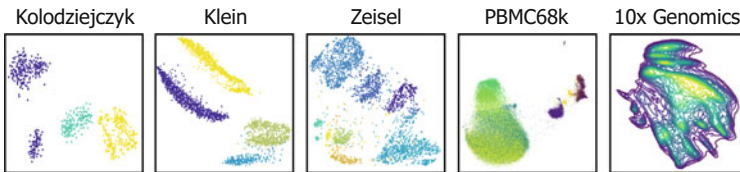


Fig. 1. Example 2D visualizations of single-cell RNA-seq datasets by net-SNE

Overall, our results demonstrate that net-SNE not only learns high quality maps like t-SNE, but also gracefully generalizes to unseen cells. This allows net-SNE to efficiently visualize mega-scale single-cell data by using a pre-trained embedding from a subsampled or an existing dataset. Our work is widely applicable to other data science domains with millions of data points to be visualized.

Acknowledgements. This work was partially supported by NIH R01GM081871.

References

1. Van Der Maaten, L.: Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **15**(1), 3221–3245 (2014)
2. 10x Genomics: Transcriptional Profiling of 1.3 Million Brain Cells with the ChromiumTM Single Cell 3' Solution. Application Note (2017). <https://www.10xgenomics.com/single-cell/>. Accessed October 2017



Probabilistic Count Matrix Factorization for Single Cell Expression Data Analysis

G. Durif^{1,2}(✉), L. Modolo^{1,3,4}, J. E. Mold⁴, S. Lambert-Lacroix⁵,
and F. Picard¹

¹ LBBE, UMR CNRS 5558, Université Lyon 1, 69622 Villeurbanne, France
ghislain.durif@inria.fr

² Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK,
38000 Grenoble, France

³ LBMC UMR 5239 CNRS/ENS Lyon, 69007 Lyon, France

⁴ Department of Cell and Molecular Biology, Karolinska Institutet,
Stockholm, Sweden

⁵ UMR 5525 Université Grenoble Alpes/CNRS/TIMC-IMAG,
38041 Grenoble, France

The combination of massive parallel sequencing with high-throughput cell biology technologies has given rise to single-cell Genomics. Similar to the paradigm shift of the 90s characterized by the first molecular profiles of tissues, it is now possible to characterize molecular heterogeneities at the cellular level (Saliba et al. 2014). The statistical characterization of heterogeneities in single-cell expression data thus requires an appropriate model, since the transcripts abundance is quantified for each cell using read counts. Hence, standard methods based on Gaussian assumptions are likely to fail to catch the biological variability of lowly expressed genes, and Poisson or Negative Binomial distributions constitute an appropriate framework (Chen et al. 2016). Moreover, dropouts, either technical (due to sampling difficulties) or biological (no expression or stochastic transcriptional activity), constitute another major source of variability in scRNA-seq (single-cell RNA-seq) data, which has motivated the development of the so-called Zero-Inflated models (Kharchenko et al. 2014). A standard and popular way of quantifying and visualizing the variability within a dataset is dimension reduction, principal component analysis (PCA) being the most widely used technique in practice. Model-based PCA (Collins et al. 2001) offers the unique advantage to be adapted to the data distribution and to be based on an appropriate metric, the Bregman divergence. It consists in specifying the distribution of the data through a statistical model. A probabilistic zero-inflated version of the Gaussian PCA was proposed by Pierson and Yau (2015) in the context of single cell data analysis (the ZIFA method). However, scRNA-seq data may be better analyzed by methods dedicated to count data such as the Non-negative Matrix Factorization (Lee and Seung 1999, NMF) or the Gamma-Poisson factor model (Cemgil 2009). However, none of the currently available dimension reduction methods fully model single-cell expression data, characterized by overdispersed zero inflated counts (Zappia et al. 2017). Our method is based on a probabilistic count matrix factorization (pCMF). We propose a dimension reduction method that is dedicated to over-dispersed counts

with dropouts, in high dimension. Our factor model takes advantage of the Poisson Gamma representation to model counts from scRNA-seq data (Zappia et al. 2017). In particular, we use Gamma priors on the distribution of principal components. We model dropouts with a Zero-Inflated Poisson distribution, and we introduce sparsity in the model thanks to a spike-and-slab approach (Malsiner-Walli and Wagner 2011) that is based on a two component sparsity-inducing prior on loadings (Titsias and Lázaro-Gredilla 2011). The model is inferred using a variational EM algorithm that scales favorably to data dimension, as compared with Markov Chain Monte Carlo (MCMC) methods (Blei et al. 2017). Then we propose a new criterion to assess the quality of fit of the model to the data, as a percentage of explained deviance, because the standard variance reduction that is used in PCA needs to be adapted to the new framework dedicated to counts. We show that pCMF better catches the variability of simulated data and experimental scRNA-seq datasets. Finally, pCMF is available in the form of a R package available at <https://gitlab.inria.fr/gdurif/pCMF>.

References

- Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* (2017). (just-accepted)
- Cemgil, A.T.: Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience* (2009)
- Chen, H.-I.H., Jin, Y., Huang, Y., Chen, Y.: Detection of high variability in gene expression from single-cell RNA-seq profiling. *BMC Genomics* **17**(Suppl 7) (2016)
- Collins, M., Dasgupta, S., Schapire, R.E.: A generalization of principal components analysis to the exponential family. In: *Advances in Neural Information Processing Systems*, pp. 617–624 (2001)
- Kharchenko, P.V., Silberstein, L., Scadden, D.T.: Bayesian approach to single-cell differential expression analysis. *Nat. Methods* **11**(7), 740 (2014)
- Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–791 (1999)
- Malsiner-Walli, G., Wagner, H.: Comparing spike and slab priors for Bayesian variable selection. *Austrian J. Stat.* **40**(4), 241–264 (2011)
- Pierson, E., Yau, C.: ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16**, 241 (2015)
- Saliba, A.-E., Westermann, A.J., Gorski, S.A., Vogel, J.: Single-cell RNA-seq: advances and future challenges. *Nucleic Acids Res.* **42**(14), 8845–8860 (2014)
- Titsias, M. K., & Lázaro-Gredilla, M.: Spike and slab variational inference for multi-task and multiple kernel learning. In: *Advances in Neural Information Processing Systems*, pp. 2339–2347 (2011)
- Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017)



Fixed-Parameter Tractable Sampling for RNA Design with Multiple Target Structures

Stefan Hammer^{1,2,3}, Yann Ponty^{4,5(✉)}, Wei Wang^{4,5}, and Sebastian Will²

¹ Department of Computer Science and Interdisciplinary Center for Bioinformatics, University Leipzig, 04107 Leipzig, Germany

² Department of Theoretical Chemistry, Faculty of Chemistry, University of Vienna, 1090 Vienna, Austria

³ Research Group Bioinformatics and Computational Biology, Faculty of Computer Science, University of Vienna, 1090 Vienna, Austria

⁴ CNRS UMR 7161 LIX, Ecole Polytechnique, Bat. Turing, 91120 Palaiseau, France

⁵ AMIBio team, Inria Saclay, Bat Alan Turing, 91120 Palaiseau, France
yann.ponty@lix.polytechnique.fr

Motivation. Engineering artificial biological systems promises broad applications in synthetic biology, biotechnology and medicine. Here, the rational design of multi-stable RNA molecules is especially powerful, since RNA can be generated with highly specific properties and programmable functions. In particular, designing artificial riboswitches became popular due to their potential as versatile biosensors [1]. Effective in-silico methods proved to greatly facilitate the design approach and have tremendous impact on their cost and feasibility.

Statement of Problem. Most methods for computational design share a similar overall strategy: one or several initial seed sequences are generated and optimized subsequently. In this contribution we revisit the first main ingredient of (multi-target) design methods, namely the sampling of sequences, which energetically favor several given target structures at the same time. While previous multi-target methods [4, 6] relied on *ad-hoc* sampling strategies, sampling seeds from the uniform distribution was solved only recently [2, 3].

Algorithmic Contributions. We generalize Boltzmann sampling for RNA design, which was recently shown powerful for single targets in *IncaRNation* [5], to design for multiple structural targets. After showing that even uniform sampling is $\#P$ -hard, we introduce the tree decomposition-based fixed parameter tractable (FPT) sampling algorithm *RNARedPrint*. Finally, we combine our FPT stochastic sampling algorithm with multi-dimensional Boltzmann sampling over distributions controlled by expressive RNA energy models. We show that sampling t sequences of length n for k target structures takes $\mathcal{O}(2^d n k + t n k)$ time, where $d := \min(w + c + 1, 2(w + 1))$, depending on the tree width w of the dependency graph (covering all dependencies between sequence positions introduced by the energy function) as well as the number c of connected components in the compatibility graph (covering the constraints enforcing canonical base pairings). Due

to a constraint framework, **RNARedPrint** supports generic Boltzmann-weighted sampling for arbitrary additive RNA energy models; this moreover enables targeting specific free energies or GC-content, compare Fig. 1.

Empirical Results. We study general properties of the approach and generate biologically relevant multi-target Boltzmann-weighted designs. Thereby, we observe significant improvements over ad-hoc methods or even uniform sampling.

Extensibility of the Approach. The presented framework is designed to enable even more general new possibilities for sequence generation in the field of RNA sequence design by enforcing additional constraints, including more complex sequence constraints, e.g. forbidden motifs in the designed sequences.

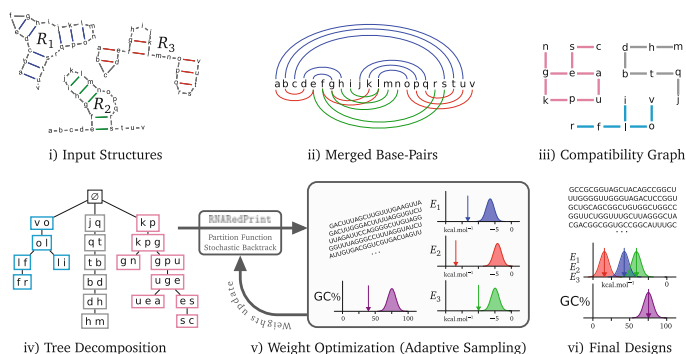


Fig. 1. General outline of **RNARedPrint**. From a set of target secondary structures (i), base-pairs are merged (ii) into a compatibility graph (iii). Based on its tree decomposition (iv), we compute the partition function, followed by a Boltzmann sampling of valid sequences (v). An adaptive scheme learns weights to achieve targeted energies and GC-content, leading to the production of suitable designs (vi).

Availability as free software: <https://github.com/yampony/RNARedPrint>

References

1. Domin, G., Findeiß, S., Wachsmuth, M., Will, S., Stadler, P.F., Mörl, M.: Applicability of a computational design approach for synthetic riboswitches. *Nucleic Acids Res.* **45**(7), 4108–4119 (2017)
2. Hammer, S., Tschischek, B., Flamm, F., Hofacker, I.L., Findeiß, S.: RNABlueprint: flexible multiple target nucleic acid sequence design. *Bioinformatics* **33**, 2850–2858 (2017)
3. Höner zu Siederdisen, C., Hammer, S., Abfalter, I., Hofacker, I.L., Flamm, C., Stadler, P.F.: Computational design of RNAs with complex energy landscapes. *Biopolymers* **99**, 1124–1136 (2013)

4. Lyngso, R.B., Anderson, J.W.J., Sizikova, E., Badugu, A., Hyland, T., Hein, J.: Frnakenstein: multiple target inverse RNA folding. *BMC Bioinf.* **13**, 260 (2012)
5. Reinharz, V., Ponty, Y., Waldispühl, J.: A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics* **29**, i308–i315 (2013)
6. Taneda, A.: Multi-objective optimization for RNA design with multiple target secondary structures. *BMC Bioinf.* **16**, 280 (2015)



Contribution of Structural Variation to Genome Structure: TAD Fusion Discovery and Ranking

Linh Huynh¹ and Fereydoon Hormozdiari^{1,2,3}(✉)

¹ Genome Center, UC Davis, Davis, USA

² MIND Institute, UC Davis, Davis, USA

³ Biochemistry and Molecular Medicine, UC Davis, Davis, USA
fhormozd@ucdavis.edu

Introduction

The significant contribution of structural variants (e.g. deletion, insertion, and inversion) to function, disease, and evolution is well reported. However, in many cases, the mechanism by which these variants contribute to the phenotype is not well understood. This is especially the case for studying non-coding structural variants and their potential biological impact. With the advent of high-throughput chromosome conformation capture (Hi-C [1]) we have novel insights into genome structure and its contribution to gene regulation. Using Hi-C data we are able to study the genomic interactions, such as enhancer-promoter interactions that are the main mechanism for gene regulation. The analysis of Hi-C data has also provided evidence that genome folds into different compartments and domains which guide the regions of the genome that can interact with each other. One of these types of domains discovered is called topological associated domains (TADs) and has provided a novel understanding of how genome structure contributes to regulation [2]. Recent studies reported structural variants (SVs) that disrupted the three-dimensional genome structure by fusing two TADs, such that enhancers from one TAD interacted with genes from the other TAD, could cause severe developmental disorders [3]. However, no method exists for directly scoring and ranking structural variations based on their effect on the three-dimensional structure such as the TAD disruption. In this paper, we formally define TAD fusion and provide a combinatorial approach for assigning a score to quantify the level of TAD fusion for each deletion denoted as TAD fusion score.

Methods

Our goal is to develop a computational method that can provide a score for deletions based on its level of modifying the 3D genomic structure and potential of causing a TAD fusion. In our method, the input consists of a Hi-C contact matrix of the genome with reference allele (i.e., without the deletion) and the

coordinates of the deletion. The output is a score representing the number of new genomic interactions made (i.e., TAD fusion score) as a result of the deletion. For this paper, we are only considering deletions, however, this approach can be extended to consider other SV types (e.g. translocations).

We propose a two-step framework for calculating the TAD fusion score: (i) predicting a new Hi-C contact matrix G of the mutated chromosome (i.e. with the deletion) given the Hi-C contact matrix H of a genome without the deletion and the deletion coordinates as the inputs; (ii) comparing this predicted/new Hi-C contact matrix G with the original Hi-C contact matrix H to estimate the number of new interactions created as a result of that deletion. For the first step, we extend the power law model (i.e. length-based model) by adding new parameters that represent the TAD structure. By that, all model parameter values can be estimated by solving a linear programming. For the second step, we define TAD fusion score as the expected number of additional genomic interactions created as a result of the deletion. Here, the genomic interactions can be defined by a simple step function or by a Bayesian formula.

Results

We show that our extended model gives a better prediction of the Hi-C contact matrix than the (length-based) power law model. In addition, our method can accurately score deletions which result in TAD fusion, and it outperforms the approaches which use predicted TADs to overlay the deletion on them for predicting TAD fusion. Furthermore, we show that our method correctly gives higher scores to deletions reported to cause developmental disorders as a result of disrupting genome structure in comparison to the deletions reported in the 1000 genomes project. Finally, we also show that deletions that cause TAD fusion are rare and under negative selection in general population.

TAD fusion score is available at <https://github.com/huynhvietlinh/FusionScore>.

References

1. Lieberman-Aiden, E., Van Berkum, N.L., Williams, L., Imakaev, M., Ragozcy, T., Telling, A., Amit, I., Lajoie, B.R., Sabo, P.J., Dorschner, M.O., et al.: Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**(5950), 289–293 (2009)
2. Dixon, J.R., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., Hu, M., Liu, J.S., Ren, B.: Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**(7398), 376–380 (2012)
3. Lupiáñez, D.G., Kraft, K., Heinrich, V., Krawitz, P., Brancati, F., Klopocki, E., Horn, D., Kayserili, H., Opitz, J.M., Laxova, R., et al.: Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* **161**(5), 1012–1025 (2015)



Assembly of Long Error-Prone Reads Using Repeat Graphs

Mikhail Kolmogorov¹(✉), Jeffrey Yuan², Yu Lin³, and Pavel Pevzner¹

¹ Department of Computer Science and Engineering, University of California,
San Diego, La Jolla, USA
mkolmogo@ucsd.edu

² Graduate Program in Bioinformatics and Systems Biology, University of California,
San Diego, La Jolla, USA

³ Research School of Computer Science, Australian National University,
Canberra, Australia

The problem of genome assembly is ultimately linked to the *repeat characterization problem*, the compact representation of all repeat families in a genome as a *repeat graph* [1]. Long read technologies have not made the repeat characterization problem irrelevant. Instead, they have simply shifted the focus from short repeats to longer repeats comparable in length to the median SMS read size; e.g., Kamath et al. [2] analyzed many bacterial genomes that existing SMS assemblers failed to assemble into a single contig. Since even bacterial (let alone, eukaryotic) genomes have long repeats, SMS assemblers currently face the same challenge that short read assemblers faced a decade ago, albeit at a different scale of repeat lengths.

Most algorithms for assembling long error-prone reads use an *overlap-layout-consensus (OLC)* approach that does not provide a repeat characterization [3, 4]. In contrast, *de Bruijn graphs* emerged as a popular approach for short read assembly because they offered an elegant representation of all repeats in a genome that reveals their mosaic structure. Most short read assemblers construct the de Bruijn graph based on all k -mers in reads and further transform it into an *assembly graph* using various *graph simplification* procedures. However, in the case of SMS reads, the key assumption of the de Bruijn graph approach (that most k -mers from the genome are preserved in multiple reads) does not hold even for short k -mers, let alone for long k -mers (e.g., $k = 1000$). As a result, various issues that have been addressed in short read assembly (e.g., how to deal with the fragmented de Bruijn graph, how to transform it into an assembly graph, etc.) remain largely unaddressed in the case of the de Bruijn graph approach to SMS assemblies.

Here, we describe the Flye algorithm for constructing repeat graphs (which have properties similar to de Bruijn graphs) from SMS reads. Flye is built on top of the ABruijn assembler [5], which generates *accurate* overlapping contigs but does not reveal the repeat structure of the genome. In contrast to ABruijn, Flye initially generates *inaccurate* overlapping contigs (i.e., contigs with potential assembly errors representing random walks on the true repeat graph) and combines these *initial* contigs into an accurate assembly graph that encodes all possible assemblies consistent with the reads. Flye further resolves *bridged* repeats

in the assembly graph thus constructing a new, less tangled assembly graph, and finally outputs accurate *final* contigs formed by paths in this graph. Flye also introduces a new algorithm that uses small differences between repeat copies to resolve *unbridged* repeats that are not spanned by any reads. We benchmarked Flye against several state-of-the-art SMS assemblers using various datasets and demonstrated that it generates accurate assemblies while also providing insight into how to plan additional experiments (e.g., using contact or optical maps) to finish the assembly. Flye is freely available at <http://github.com/fenderglass/Flye>.

References

1. Pevzner, P.A., Tang, H., Tesler, G.: De novo repeat classification and fragment assembly. *Genome Res.* **14**(9), 1786–1796 (2004)
2. Kamath, G.M., Shomorony, I., Xia, F., Courtade, T.A., Tse, N.: D: HINGE: long-read assembly achieves optimal repeat resolution. *Genome Res.* **27**(5), 747–756 (2017)
3. Chin, C.S., Peluso, P., Sedlazeck, F.J., Nattestad, M., Concepcion, G.T., Clum, A., Dunn, C., O'Malley, R., Figueroa-Balderas, R., Morales-Cruz, A., Cramer, G.R.: Phased diploid genome assembly with single-molecule real-time sequencing. *Nat. Methods* **13**(12), 1050 (2016)
4. Koren, S., Walenz, B.P., Berlin, K., Miller, J.R., Bergman, N.H., Phillippy, A.M.: Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* **27**(5), 722–736 (2017)
5. Lin, Y., Yuan, J., Kolmogorov, M., Shen, M.W., Chaisson, M., Pevzner, P.A.: Assembly of long error-prone reads using de Bruijn graphs. *Proc. Nat. Acad. Sci.* **113**(52), E8396–E8405 (2016)



A Multi-species Functional Embedding Integrating Sequence and Network Structure

Mark D. M. Leiserson¹, Jason Fan¹, Anthony Cannistra², Inbar Fried³,
Tim Lim⁴, Thomas Schaffner⁵, Mark Crovella⁴, and Benjamin Hescott⁶(✉)

¹ Department of Computer Science, University of Maryland, College Park, USA

² Department of Biology, University of Washington, Seattle, USA

³ University of North Carolina Medical School, Chapel Hill, USA

⁴ Department of Computer Science, Boston University, Boston, USA

⁵ Department of Computer Science, Princeton University, Princeton, USA

⁶ College of Computer and Information Science, Northeastern University,
Boston, USA

b.hescott@northeastern.edu

Introduction. Transferring biological knowledge between species is fundamental for many important problems in genetics. These problems range from the molecular-level, such as predicting protein function or genetic interactions [4], to the organism-level, such as predicting human disease models [5]. The most common approach researchers have taken is to use orthologs inferred from DNA sequencing data. More recently, researchers have sought to expand beyond sequence-based orthologs using high-throughput proteomics data under the hypothesis that genes with similar topology in protein-protein interaction (PPI) networks have similar functions. Many methods have been introduced to infer homology across species (i.e. a node matching) from sequence similarity and PPI networks, including network alignment [1]. More recently, Jacunski, et al. [4] identified *connectivity homologous* gene pairs using a small set of features derived from PPI networks. These prior works are focused on node matching and constructing node feature vectors, but do not address the problem of embedding genes from different species into a shared, general-purpose space.

Methods. We introduce a new algorithm, Homology Assessment across Networks using Diffusion and Landmarks (HANDL), that leverages graph kernels to embed nodes from two PPI networks into a biologically meaningful and general-purpose vector space using network and sequence data.¹ Kernels, particularly kernels that capture random walks and/or heat diffusion processes on graphs, have been widely and successfully used for computing similarity between nodes within biological networks [2].

The main computational challenge HANDL solves is relating network kernel matrices from different species. Because the kernel matrices from networks of

¹ An implementation of HANDL is available at <https://github.com/lrgr/HANDL>.

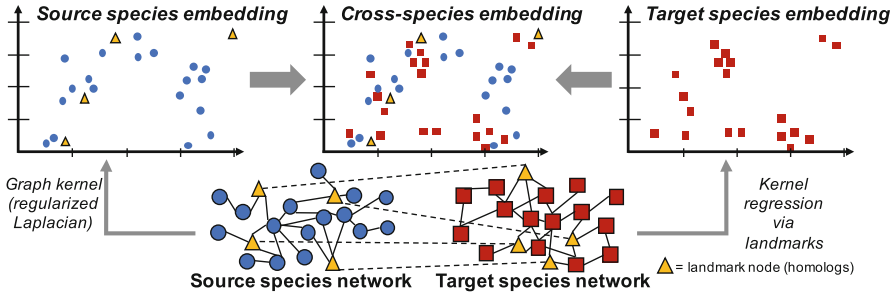


Fig. 1. HANDL embeds nodes into a shared vector space.

different species have different dimensions, traditional kernel transfer learning approaches (e.g. [3]) cannot be directly applied. We show a schematic of the HANDL algorithm in Fig. 1. HANDL takes as input a *source* network, a *target* network, and a set of *landmarks* shared between the networks to embed nodes from the target species into the vector space of the source species. The inner-product between embeddings gives *HANDL similarity scores* between nodes in different species. As HANDL is a general algorithm, the landmarks and graph kernel can be customized for particular applications. In this work, we use a subset of homologs between the source and target species as landmarks and the regularized Laplacian kernel specifically to capture protein functional similarity.

Results. We show that the human-mouse and baker’s-fission yeast cross-species embeddings constructed by HANDL are biologically meaningful with three cross-species tasks. First, we find that HANDL similarity scores are strongly correlated with cross-species functional similarity, and that pairs with the highest HANDL similarity scores are more functionally similar than pairs with the closest connectivity homology profiles [4]. Next, we use the algorithm and data from McGary, et al. [5] and *HANDL-homologs* (node pairs with high HANDL similarity scores) to find new, novel human-mouse disease models (phenologs, i.e. orthologous phenotypes) that are supported by biological literature. Finally, we show that node vectors themselves are of more general use. We use HANDL to transfer knowledge of synthetic lethal (SL) interactions in baker’s to fission yeast (and vice versa). We compute embeddings for the source and target species then train a support vector machine (SVM) only on embeddings of the source species. We find that that the SVM also separates embeddings of the target species with respect to SLs and non-SLs on previously unseen data.

These results show how HANDL can transfer knowledge of genetics between humans and model organisms. We anticipate that HANDL can serve as the foundation for more sophisticated approaches for transfer learning across species.

References

1. Clark, C., Kalita, J.: A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics* **30**(16), 2351–2359 (2014)
2. Cowen, L., Ideker, T., Raphael, B.J., Sharan, R.: Network propagation: a universal amplifier of genetic associations. *Nat. Rev. Genet.* (2017)
3. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K. M., and Scholkopf, B.: Correcting sample selection bias by unlabeled data. *Adv. Neural Inf. Process. Syst.* 601–608 (2006)
4. Jacunski, A., Dixon, S.J., Tatonetti, N.P.: Connectivity homology enables inter-species network models of synthetic lethality. *PLoS Comp. Bio.* **11**(10), e1004506 (2015)
5. McGary, K.L., Park, T., Woods, J.O., Cha, H., Wallingford, J.B., Marcotte, E.M.: Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proc. Natl. Acad. Sci.* 107(14), 6544–6549 (2010)



Deciphering Signaling Specificity with Deep Neural Networks

Yunan Luo¹, Jianzhu Ma², Yang Liu¹, Qing Ye¹, Trey Ideker²,
and Jian Peng¹(✉)

¹ Department of Computer Science, University of Illinois at Urbana-Champaign,
Champaign, USA

² School of Medicine, University of California San Diego, La Jolla, USA
jianpeng@illinois.edu

1 Introduction

Protein kinase phosphorylation is one of the primary forms of post-translation modification (PTM) that transduce cellular signals and regulate cellular processes. Defective signal transductions, which are associated with protein phosphorylation, have been linked to many human diseases, such as cancer. Defining the organization of the phosphorylation-based signaling network and, in particular, identifying kinase-specific substrates can help reveal the molecular mechanism of the signaling network and understand their impacts on human diseases.

2 Methods

We present DeepSignal, a deep learning based method for predicting the substrate specificity of kinase domains. Unlike most of the previous methods that only focus on using substrate sequences to derive the kinases specificity, DeepSignal takes into account the information in both kinase domain sequences and substrate peptides, and translates a kinase sequences into its specificity profile (e.g., a position-specific scoring matrix, PSSM). DeepSignal employs the Long Short-Term Memory (LSTM) network, a deep learning architecture with memory units, to process the kinase sequences with various lengths using a single model, enabling the learning of universal knowledge across multiple kinase domains. Our deep learning based method is able to automatically extract complex features in kinase domain sequences that best explains the substrate specificity of this kinase. For example, with the memory ability of LSTM, DeepSignal can exploit and record the long and short range dependencies between residues spanning over an arbitrary distance in the kinase domain, which is challenging for previous non-deep learning methods of phosphosites prediction. In addition, DeepSignal can transfer the knowledge from currently available kinase-substrate data to predict phosphosites for new kinases, which is infeasible for many existing kinase-specific methods.

Y. Luo, J. Ma, and Y. Liu — Equal contribution.

3 Results

We evaluated the ability of DeepSignal on predicting the substrate specificity of kinase domains. Our method is able to achieve 0.875 AUROC (area under the receiver operating characteristic curve) and 0.21 AUPRC (area under the precision-recall curve) scores in a five-fold cross-validation, which is a substantial improvement over previous methods GPS 2.0 [1] and NetPhorest [2]. To test the generalization ability of our method, we further apply DeepSignal to predict the binding specificity of SH2 domain (Fig. 1), another phosphorylation-based signaling modular domain, on four high-throughput datasets. DeepSignal significantly outperforms two SH2-peptide interaction methods (SMALI [3] and SH2PepInt [4]) and one general protein-protein interaction method (PrePPI [5]). Although trained on 80% of the data in the five-fold cross-validation, our method still achieves higher or comparable AUROC scores when compared to a method (MSM/D-PEM [6]) that was pre-trained on all the binding data of each dataset. Overall, these results demonstrated the ability of DeepSignal on predicting the binding specificity of phosphorylation-based signaling domains.

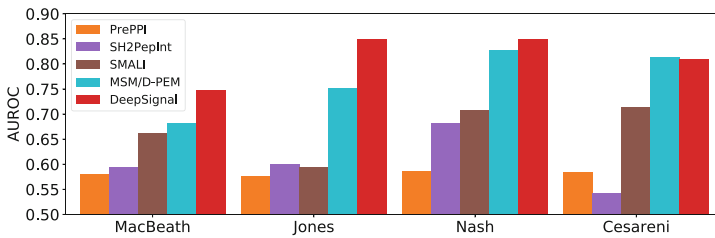


Fig. 1. Evaluation of prediction performance on prediction of the binding between SH2 domains and phosphotyrosine peptides.

To study the impact of mutations on cancer, we used DeepSignal to construct the signaling network using only the protein primary sequences of 16,254 proteins, including 307 kinase domains, 122 SH2 domains and 190,427 phosphoproteins across 18 cancer types. For each cancer type, we mapped all the coding mutations from TCGA on the protein sequences. This resulted 6,286 mutations on kinase domains, 776 mutations on SH2 domains and 37,996 mutations on phosphoproteins. We use DeepSignal to quantify the change of the binding specificity caused by the cancer mutations of a given kinase/SH2-peptide, and predict a ranking list of single-nucleotide variants (SNV) that potentially disrupt phosphosites. We found DeepSignal is more sensitive in detecting known cancer genes related to signaling transduction than an existing statistical approach [6]. DeepSignal can further discover new perturbed pathways related to cancer including CTNNB1 pathway in UCEC, PTEN pathway in GBM and SMAD4 pathway in LUAD.

Acknowledgments. This work was supported in part by the NSF CAREER Award, the Sloan Research Fellowship, and the PhRMA Foundation Award in Informatics.

References

1. Xue, Y., et al.: Gps 2.0, a tool to predict kinase-specific phosphorylation sites in hierarchy. *Mol. Cell. Proteomics* **7**, 1598–1608 (2008)
2. Miller, M.L., et al.: Linear motif atlas for phosphorylation-dependent signaling. *Sci. Signal* **1**, ra2 (2008)
3. Li, L., et al.: Prediction of phosphotyrosine signaling networks using a scoring matrix-assisted ligand identification approach. *Nucleic Acids Res.* **36**, 3263–3273 (2008)
4. Kundu, K., Costa, F., Huber, M., Reth, M., Backofen, R.: Semi-supervised prediction of sh2-peptide interactions from imbalanced high-throughput data. *PloS one* **8**, e62732 (2013)
5. Zhang, Q.C., Petrey, D., Garzón, J.I., Deng, L., Honig, B.: Preppi: a structure-informed database of protein-protein interactions. *Nucleic Acids Res.* **41**, D828–D833 (2012)
6. AlQuraishi, M., Koytiger, G., Jenney, A., MacBeath, G., Sorger, P.K.: A multi-scale statistical mechanical framework integrates biophysical and genomic data to assemble cancer networks. *Nat. Genet.* **46**, 1363–1371 (2014)



Integrative Inference of Subclonal Tumour Evolution from Single-Cell and Bulk Sequencing Data

Salem Malikic¹, Katharina Jahn^{2,3}, Jack Kuipers^{2,3}, S. Cenk Sahinalp⁴(✉),
and Niko Beerenwinkel^{2,3}(✉)

¹ School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

² Department of Biosystems Science and Engineering, ETH Zurich,
Basel, Switzerland

`niko.beerenwinkel@bsse.ethz.ch`

³ SIB Swiss Institute of Bioinformatics, Basel, Switzerland

⁴ Department of Computer Science, Indiana University, Bloomington, IN, USA
`cenksahi@indiana.edu`

Cancer is a genetic disease that develops through a branched evolutionary process. It is characterised by the emergence of genetically distinct subclones through the random acquisition of mutations at the level of single-cells and shifting prevalences at the subclone level through selective advantages purveyed by driver mutations. This interplay creates complex mixtures of tumour cell populations which exhibit different susceptibility to targeted cancer therapies and are suspected to be the cause of treatment failure. Therefore it is of great interest to obtain a better understanding of the evolutionary histories of individual tumours and their subclonal composition.

Most of the current data on tumour genetics stems from short read bulk sequencing data. While this type of data is characterised by low sequencing noise and cost, it consists of aggregate measurements across a large number of cells. It is therefore of limited use for the accurate detection of the distinct cellular populations present in a tumour and the unambiguous inference of their evolutionary relationships. Single-cell DNA sequencing instead provides data of the highest resolution for studying intra-tumour heterogeneity and evolution, but is characterised by higher sequencing costs and elevated noise rates.

As the strengths and weaknesses of bulk and single-cell sequencing data are to a large extent complimentary with respect to phylogeny inference, using both data types for a joint inference should improve our understanding of subclonal tumour evolution over using each type of data alone. In this work, we develop B-SCITE, the first computational approach that infers trees of tumour evolution from combined bulk and single-cell sequencing data. B-SCITE employs an MCMC search scheme to find the mutation tree that maximizes the joint likelihood of both data types. The model accounts for typical sequencing biases and artifacts, including the variability in depth of coverage among different bulk sequencing datasets and the contamination of single-cell data by doublets. Using

S. Malikic, K. Jahn, and J. Kuipers — Equal contributors.

a comprehensive set of simulated data, we show that B-SCITE systematically outperforms existing methods with respect to tree reconstruction accuracy and subclone identification. High-fidelity reconstructions are obtained even with a modest number of single cells, suggesting that combined bulk and single-cell data may be a competitive strategy for tumor phylogeny reconstruction. On real data, we show that B-SCITE provides more realistic mutation histories compared to the results reported in previous studies or obtained by existing methods.



Mantis: A Fast, Small, and Exact Large-Scale Sequence-Search Index

Prashant Pandey¹(✉), Fatemeh Almodaresi¹, Michael A. Bender¹,
Michael Ferdman¹, Rob Johnson^{1,2}, and Rob Patro¹

¹ Computer Science Department, Stony Brook University, Stony Brook, USA
{ppandey, falmodaresit, bender, mferdman, rob.patro}@cs.stonybrook.edu,
robj@vmware.com

² VMware Research, Palo Alto, USA

The ability to issue sequence-level searches over publicly available databases of assembled genomes and known proteins has played an instrumental role in many studies in the field of genomics, and has made BLAST [2] and its variants some of the most widely-used tools in all of science. However, until recently, tools for searches over genomic data were restricted to reference sequences. As a result, the vast majority of publicly-available sequencing data (e.g., the data deposited in the SRA [3]) has been difficult to search because it exists in the form of raw, unassembled sequencing reads.

Recently, Solomon and Kingsford introduced the sequence Bloom tree (SBT) [8] for performing searches over thousands of sequencing experiments. This seminal work introduced both a formulation of this problem, and the initial steps toward a solution. The space and query time of the SBT structure has been further improved by Solomon and Kingsford [9] and Sun et al. [10].

Sequence Bloom trees repurpose Bloom filters to index large sets of raw sequencing data probabilistically and, as a result, they are forced to cope with Bloom filters' limitations. For example, the SBT needs to merge Bloom filters, but Bloom filters must be the same size to be merged, and they cannot be resized. Consequently, SBTs use Bloom filters of the same size to represent sets of widely varying cardinalities. As a result, most of the Bloom filters in the SBT are sub-optimally tuned and inefficient in their use of space. (SBTs partially mitigate this issue by compressing their Bloom filters using an off-the-shelf compressor.)

We introduce Mantis, a space-efficient data structure that can be used to index thousands of raw-read experiments and facilitate large-scale sequence searches on those experiments. Mantis uses counting quotient filters [5] instead of Bloom filters, enabling rapid index builds and queries, small indexes, and *exact* results, i.e., no false positives or negatives. Furthermore, Mantis is also a colored De Bruijn graph (cDBG) representation, and supports the same fast de Bruijn graph traversals as Squeakr [4], and hence may be useful for topological analyses such as computing the length of the query covered in each experiment (rather than just the fraction of k -mers present).

Mantis has several advantages over prior work:

- Mantis is *exact*. A query for a set Q of k -mers and threshold θ returns exactly those data sets containing at least fraction θ of the k -mers in Q . There are no false positives or false negatives. In contrast, we show that SBT-based systems exhibit only 57–67% precision, meaning that many of the results returned for a given query are, in fact, false positives.
- Mantis supports much faster queries than existing SBT-based systems. In our experiments, queries in Mantis ran up to 100× faster than when using an (in RAM) SSBT.
- Mantis supports much faster index construction. For example, we were able to build the Mantis index on 2,652 data sets in 16 hours and 35 min. SSBT reported 97 hours to construct an index on the same collection of data sets.
- Mantis uses less storage than SBT-based systems. For example, the Mantis index over the 2,652 experiments used for evaluation is 20% smaller than the compressed SSBT index.
- Mantis returns, for each experiment containing at least 1 k -mer from the query, the number of query k -mers present in this experiment. Thus, the full spectrum of relevant experiments can be analyzed. While these results can be post-processed to filter out those not satisfying a θ -query, we believe the Mantis output is more useful, as one can analyze which experiments were close to achieving the θ threshold, and can examine if a natural filtering “cutoff” exists.

Mantis builds on Squeakr, a k -mer counter based on the counting quotient filter (CQF). Prior work has shown how CQFs can be used to improve performance and simplify the design of k -mer-counting tools [4] and de Bruijn graph representations [6].

In a similar spirit, Mantis uses the CQF to create a simple space- and time-efficient index for searching for sequences in large collections of experiments. Mantis is based on cDBGs. The “color” associated with each k -mer in a cDBG is the set of experiments in which that k -mer occurs (similar to Rainbowfish [1]). We use an exact CQF to store a table mapping each k -mer to a color ID, and another table mapping color IDs to the actual set of experiments containing that k -mer. Mantis uses an off-the-shelf compressor [7] to store the bit vectors representing each set of experiments.

Mantis takes as input the collection of CQFs representing each data set, and outputs the search index. Construction is efficient because it can use sequential I/O to read the input and write the output CQFs. Similarly, queries for the color of a single k -mer are efficient since they require only two table lookups.

Mantis is available at <https://github.com/splatlab/mantis>.

References

1. Almodaresi, F., Pandey, P., Patro, R.: Rainbowfish: A Succinct Colored de Bruijn Graph Representation. In WABI, volume 88, pages 18:1–18:15, 2017
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W.: Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990)
3. Kodama, Y., Shumway, M., Leinonen, R.: The sequence read archive: explosive growth of sequencing data. *Nucleic Acids Res.* **40**(D1), D54–D56 (2011)
4. Prashant Pandey, Michael A Bender, Rob Johnson, and Rob Patro. Squeakr: An Exact and Approximate k-mer Counting System. *Bioinformatics*, page btx636, 2017
5. Prashant Pandey, Michael A. Bender, Rob Johnson, and Robert Patro. A General-Purpose Counting Filter: Making Every Bit Count. In SIGMOD, pages 775–787, 2017
6. Prashant Pandey, Michael A. Bender, Rob Johnson, and Robert Patro. deBGR: an efficient and near-exact representation of the weighted de Bruijn graph. *Bioinformatics*, 33(14), 2017
7. Rajeev Raman, Venkatesh Raman, and S. Srinivasa Rao. Succinct indexable dictionaries with applications to encoding k-ary trees and multisets. In SODA, pages 233–242, 2002
8. Solomon, B., Kingsford, C.: Fast search of thousands of short-read sequencing experiments. *Nat. Biotechnol.* **34**(3), 300–302 (2016)
9. Solomon, B., Kingsford, C.: Improved Search of Large Transcriptomic Sequencing Databases Using Split Sequence Bloom Trees. In: Sahinalp, S.C. (ed.) RECOMB 2017. LNCS, vol. 10229, pp. 257–271. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56970-3_16
10. Sun, C., Harris, R.S., Chikhi, R., Medvedev, P.: AllSome Sequence Bloom Trees. In: Sahinalp, S.C. (ed.) RECOMB 2017. LNCS, vol. 10229, pp. 272–286. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56970-3_17



Tensor Composition Analysis Detects Cell-Type Specific Associations in Epigenetic Studies

Elior Rahmani¹(✉), Regev Schweiger², Saharon Rosset³,
Sriram Sankararaman¹, and Eran Halperin^{1,4,5}

¹ Department of Computer Science, UCLA, Los Angeles, CA, USA
elior.rahmani@gmail.com

² Blavatnik School of Computer Science, Tel-Aviv University, Tel Aviv, Israel

³ Department of Statistics, Tel Aviv University, Tel Aviv, Israel

⁴ Department of Human Genetics, UCLA, Los Angeles, CA, USA

⁵ Department of Anesthesiology and Perioperative Medicine,
UCLA, Los Angeles, CA, USA
ehalperin@cs.ucla.edu

Abstract. Identifying cell-type specific associations of genes with disease and mapping known associations to particular cell types is a key in understanding disease etiology. While developments in technologies for profiling genomic features such as gene expression and DNA methylation have led to the availability of large-scale tissue-specific genomic data, prohibitive costs drastically restrict collection of cell-type specific genomic data. This, in turn, limits the identification of disease-related genes and cell types. It is therefore desired to develop new approaches for detecting cell-type specific associations between phenotypes and tissue-specific genomic data.

We suggest a new matrix factorization formulation, which allows us to deconvolve a two-dimensional input (observations by features) into a three-dimensional output. Traditional matrix factorization formulations essentially take as an input a multiple-source heterogeneous matrix of observations and output a matrix of source-specific weights and a matrix of source-specific features. We generalize this approach by assuming that source-specific features are unique for each observation rather than shared across all observations, and we propose Tensor Composition Analysis (TCA), a method for estimating observation- and source-specific values based on the model.

We apply our model in the context of epigenetic association studies, where DNA methylation data measured from a heterogeneous tissue are often used, and we show that TCA allows us to extract cell-type specific methylation levels from two dimensional tissue-specific methylation data. We further derive a statistical test for detecting cell-type specific effects of methylation on phenotypes based on the TCA model, and using a simulation study we demonstrate its potentials and limitations. Finally, using five large whole-blood methylation datasets, we demonstrate that our model allows the detection of novel replicating cell-type specific associations without collecting cost prohibitive cell-type specific data, thus

suggesting an exciting new opportunity to unveil more of the hidden signals in genomic association studies with potential design implications for future data collection efforts.



Assembly-Free and Alignment-Free Sample Identification Using Genome Skims

Shahab Sarmashghi¹(✉), Kristine Bohmann^{2,3}, M. Thomas P. Gilbert^{2,4},
Vineet Bafna⁵, and Siavash Mirarab¹

¹ Department of Electrical and Computer Engineering,
University of California, San Diego, La Jolla, CA 92093, USA
ssarmash@ucsd.edu

² Evolutionary Genomics, Natural History Museum of Denmark,
University of Copenhagen, Copenhagen, Denmark

³ School of Biological Sciences, University of East Anglia, Norwich, Norfolk, UK

⁴ Norwegian University of Science and Technology,
University Museum, 7491 Trondheim, Norway

⁵ Department of Computer Science and Engineering,
University of California, San Diego, La Jolla, CA 92093, USA

Extended abstract

The ability to quickly and inexpensively describe the taxonomic diversity in an environment is critical in this era of rapid climate and biodiversity changes. The currently preferred molecular technique, barcoding, is low-cost and widely used, but has drawbacks. As sequencing costs continue to fall, an alternative approach based on *genome-skimming* has been proposed [1, 2]. This approach first applies low-pass (100 Mb – several Gb per sample) sequencing to voucher and/or query samples and then recovers marker genes and/or organelle genomes computationally. In contrast, we suggest the use of the unassembled sequence data for taxonomic identification using an alignment-free approach based on the k-mer decomposition of the sequencing reads. Specifically, we first estimate the average sequencing depth and error rate for each genome skim, by comparing our derived theoretical distribution of k-mers' multiplicity and the histogram of k-mer counts computed using Jellyfish [3]. The genome length is also estimated from the average sequencing depth accordingly. Then, the similarity of two genome skims is measured by the Jaccard index between their corresponding k-mer collections. Finally, the hamming distance between genomes is estimated from the Jaccard index, using the following formula obtained by modeling the impact of low sequencing coverage, sequencing error, and differing genome lengths on the similarity of genome skims:

$$D = 1 - \left(\frac{2(\zeta_1 L_1 + \zeta_2 L_2)J}{\eta_1 \eta_2 (L_1 + L_2)(1 + J)} \right)^{1/k} .$$

In this equation, when coverage is low, we use all k-mers and set:

$$\eta_i = 1 - e^{-c_i(1-k/\ell)(1-\epsilon_i)^k}, \quad \zeta_i = \eta_i + c_i(1-k/\ell)(1-(1-\epsilon_i)^k).$$

For higher coverages, we remove k-mers with multiplicity below a threshold m , and set:

$$\zeta_i = \eta_i = 1 - \sum_{t=0}^{m-1} \frac{(c_i(1-k/\ell)(1-\epsilon_i)^k)^t}{t!} e^{-c_i(1-k/\ell)(1-\epsilon_i)^k}.$$

In these equations, k and ℓ are k-mer and read length, respectively, and c_i , ϵ_i , and L_i are substituted from the estimates of coverage, error rate, and genome length for each genome skim. The Jaccard index between two genome skims, J , is computed by Mash [4] efficiently using a hashing technique.

We have tested our tool, Skmer, on genome skims simulated from assemblies of 90 species from two genera of insects (Anopheles and Drosophila) and across the avian tree of life. We test the accuracy of the distances computed by Skmer, and subsequently use the distances to find the exact/closest match to a query sample in a reference set of genome skims. Comparing to the other k-mer based tools, Skmer shows excellent performance in our simulation studies, especially when the coverage is below 4X [5].

Skmer makes the assembly-free approach to genome-skimming a viable alternative to the traditional barcoding. The software is made publicly available on Github (<https://github.com/shahab-sarmashghi/Skmer.git>).

References

1. Straub, S.C.K., Parks, M., Weitemier, K., Fishbein, M., Cronn, R.C., Liston, A.: Navigating the tip of the genomic iceberg: next-generation sequencing for plant systematics. *Am. J. Bot.* **99**(2), 349–364 (2012)
2. Coissac, E., Hollingsworth, P.M., Lavergne, S., Taberlet, P.: From barcodes to genomes: extending the concept of dna barcoding. *Mol. Ecol.* **25**(7), 1423–1428 (2016)
3. Marçais, G., Kingsford, C.: A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* **27**(6), 764–770 (2011)
4. Ondov, B.D., Treangen, T.J., Melsted, P., Mallonee, A.B., Bergman, N.H., Koren, S., Phillippy, A.M.: Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.* **17**(1), 132 (2016)
5. Sarmashghi, S., Bohmann, K., Gilbert, M.T.P., Bafna, V., Mirarab, S.: Assembly-free and alignment-free sample identification using genome skims (2017). bioRxiv 230409



Efficient Algorithms to Discover Alterations with Complementary Functional Association in Cancer

Rebecca Sarto Basso¹, Dorit S. Hochbaum¹, and Fabio Vandin^{2,3}(✉)

¹ University of California at Berkeley, Berkeley, USA
rebeccasarto@berkeley.edu, hochbaum@ieor.berkeley.edu

² University of Padova, Padova, Italy
fabio.vandin@unipd.it

³ Brown University, Providence, USA

Introduction. Recent advances in sequencing technologies now allow to assay the entire complement of somatic alterations in large tumour cohorts [5]. Several computational methods have been recently designed to identify *driver* alterations, associated to the disease, and to distinguish them from *passenger* alterations not related with the disease. The identification of driver alterations is complicated by the extensive *intertumour heterogeneity*, with large (100–1000's) and different collections of alterations being present in tumours from different patients and no two tumours having the same collection of alterations [6, 7]. One of the reasons for such heterogeneity is that driver alterations target cancer *pathways*, groups of interacting genes performing given functions in the cell and whose alteration is required to develop the disease [2, 7]. One of the main remaining challenges is the identification of alterations with functional impact [3].

Several methods for the *de novo* discovery of mutated cancer pathways have leveraged the *mutual exclusivity* of cancer alterations, with cancer pathways displaying at most one alteration for each patient [3, 7]. The mutual exclusivity property is due to the complementarity of genes in the same pathway, with alterations in different members of a pathway resulting in a similar impact at the functional level. An additional source of information that can be used to identify genes with complementary functions are quantitative measures for each samples such as functional profiles, obtained for example by genomic or chemical perturbations [1]. The employment of such quantitative measurements is crucial to identify meaningful complementary alterations since one can expect mutual exclusivity to reflect in functional properties of altered samples which are specific to the altered samples.

Methods and Results. We study the problem of finding sets of alterations with complementary functional associations using alteration data and a quantitative (functional) target measure from a collection of cancer samples. We provide a rigorous combinatorial formulation for the problem and prove that the associated computational problem is NP-hard. We develop two efficient algorithms, a greedy algorithm and an ILP-based algorithm to identify the set of k genes with the highest association with a target and prove rigorous guarantees in the quality of their solutions.

Our algorithms are implemented in our tool `fUNCTIONAL Complementary of alteratiOns discoVERY (UNCOVER)`¹. We compared UNCOVER with REVEALER [4], a recently developed greedy algorithm to identify mutually exclusive sets of alterations associated with functional phenotypes. Considering four cancer datasets from [4], we compared the solutions obtained by our algorithms with the solutions from REVEALER in terms of the *information coefficient* (IC), the target association score used in [4] as a quality of the solution. Surprisingly, in two out of four datasets our methods, which do not consider the IC score, identify solutions with IC score *higher* (by at least 5%) than the solutions reported by REVEALER, while for the other two datasets the IC score is very similar. These results show that UNCOVER identifies better solutions than REVEALER when evaluated using our objective function *and* also when evaluated according to the objective function of REVEALER.

In addition, UNCOVER has a running time that is on average two orders of magnitude smaller than required by REVEALER. The efficiency of UNCOVER enables the analysis of a large number of targets. We have run UNCOVER on a dataset with thousands of functional targets and tens of thousands alterations from the Achilles project dataset² and the Cancer Cell Line Encyclopedia (CCLE). While running UNCOVER (including preprocessing) on the entire dataset required 24 h, based on the runtime required on the instances reported in [4] running REVEALER on this dataset would have required about 5 months of compute time. On such large dataset, UNCOVER identifies several statistically significant associations between target values and mutually exclusive alterations in genes sets.

Acknowledgement. This work is supported, in part, by NSF grant IIS-124758 and by the University of Padova grants SID2017 and PROACTIVE2017. This work was done in part while FV was visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation.

References

1. Cowley, et al.: Parallel genome-scale loss of function screens in 216 cancer cell lines for the identification of context-specific genetic dependencies. *Sci Data* (2014)
2. Creixell, et al.: Pathway and network analysis of cancer genomes. *Nat. Met.* (2015)
3. Garraway and Lander: Lessons from the cancer genome. *Cell* (2013)
4. Kim, et al.: Characterizing genomic alterations in cancer by complementary functional associations. *Nat. Biotech.* (2016)
5. TCGA Research Network: Integrated genomic characterization of pancreatic ductal adenocarcinoma. *Cancer Cell* (2017)
6. Vandin: Computational methods for characterizing cancer mutational heterogeneity. *Frontiers in genetics* (2017)
7. Vogelstein, et al.: Cancer genome landscapes. *Science* (2013)

¹ <https://github.com/VandinLab/UNCOVER>.

² <https://portals.broadinstitute.org/achilles>.



Latent Variable Model for Aligning Barcoded Short-Reads Improves Downstream Analyses

Ariya Shajii¹, Ibrahim Numanagić^{1,2}, and Bonnie Berger^{1,2}(✉)

¹ Computer Science and AI Lab, MIT, Cambridge, MA, USA
bab@mit.edu

² Department of Mathematics, MIT, Cambridge, MA, USA

Background: Barcoded read sequencing allows short-reads to carry long-range information by virtue of read “barcodes”, and has several advantages (including significantly reduced cost and lower error rates) over long-read sequencing. Here we introduce a two-tiered statistical binning approach, EMerAld—or EMA for short—to barcoded read sequence alignment, an essential component of any barcoded sequencing pipeline, and as a result improve downstream genotyping and phasing. Our method enables the probabilistic placement of reads between different read clouds [1], and also in a single cloud that spans homologous elements. The two tiers consist of: (i) a novel latent variable model to probabilistically assign reads to possible source fragments; and (ii) newly exploiting expected read coverage (read density) to resolve the difficult case of multiple repetitive alignments of reads within a single read cloud. These ambiguous alignments account for a large fraction of the rare variants that currently cannot be resolved and are of great interest to biologists [2].

Methods: Current linked-read alignment methods first perform a standard all-mapping, then partition the resulting alignments into groups of nearby reads with a common barcode called “read clouds”. Reads are then assigned to one of their possible clouds by optimizing a global score function that takes into account edit distance, mate pairs, read clouds, etc. Our two main conceptual advances are as follows. Intuitively, rather than assigning each read to just one of its possible alignments at any given time, we make use of probabilistic assignments of reads to clouds and employ a latent variable model to determine final alignment probabilities; thereby, we select the most likely cloud (and thus alignment) for each read. During the cloud alignment process, we also utilize a disjoint-set data structure over read clouds to normalize alignment probabilities in a physically sensible way. Once reads are assigned to clouds, we propose a different statistical binning optimization approach to better handle the ubiquitous repetitive regions of the genome. Whereas currently-used methods simply pick the lowest edit distance alignment of a read in a given cloud, we instead optimize a combination of edit distance and “read density”, which takes into account the read density distribution over fragments. This two-tiered process can be interpreted

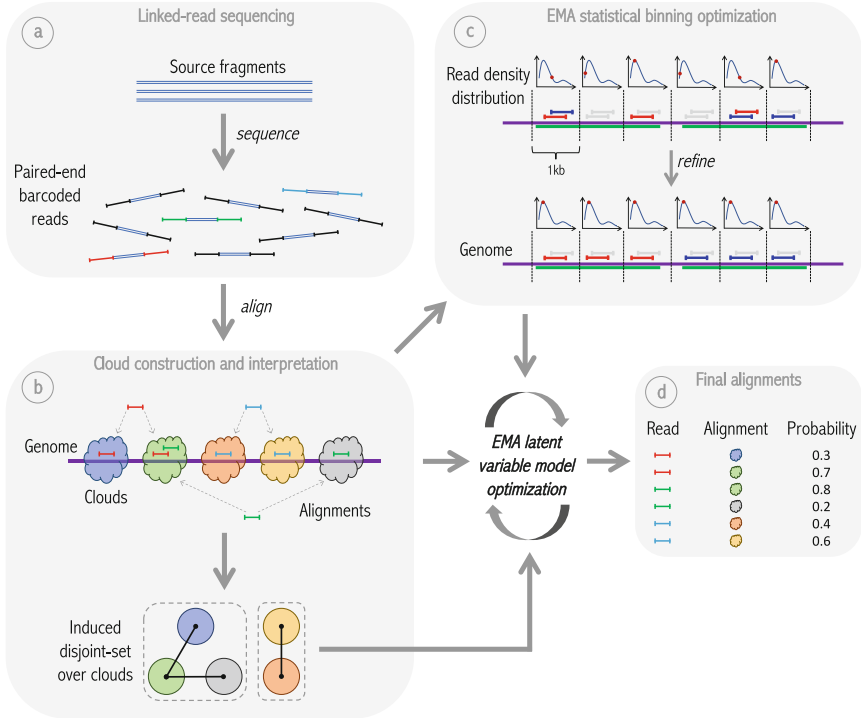


Fig. 1. Overview of EMA pipeline. **(a)** Idealized model of linked-read sequencing, wherein some number of unknown source fragments in a single droplet are sheared, barcoded and sequenced to produce linked-reads. **(b)** EMA’s “read clouds” are constructed by grouping nearby-mapping reads sharing the same barcode; these clouds represent possible source fragments. EMA then partitions the clouds into a disjoint-set induced by the alignments, where two clouds are connected if there is a read aligning to both; connected components in this disjoint-set (enclosed by dashed boxes) correspond to alternate possibilities for the *same* unknown source fragment. EMA’s latent variable model optimization is subsequently applied to each of these connected components individually. **(c)** EMA applies a novel statistical binning optimization algorithm to clouds containing multiple alignments of the same read to pick out the most likely alignment, by optimizing a combination of alignment edit distances and read densities within the cloud. In the figure, the green regions of the genome are homologous, thereby resulting in multi-mappings within a single cloud. **(d)** While the statistical binning optimization operates within a single cloud, EMA’s latent variable model optimization determines the best alignment of a given read between different clouds, and produces not only the final alignment for each read, but also interpretable alignment *probabilities*.

as statistical binning first in assigning reads to clouds and then within clouds. The EMA pipeline is shown in Fig. 1.

Results: EMA is much faster and less memory intensive compared to other tools. EMA’s overhead over the initial run of an all-mapper is virtually negligible,

and EMA is at least $1.5\times$ faster than Lariat (the current 10x alignment tool [1]), which translates into days faster for the user. In addition, we show that genotypes called from EMA’s alignments contain over 30% fewer false positives than those called from Lariat’s, with a fewer number of false negatives, on 10x WGS datasets of NA12878 and NA24385, as compared to NIST GIAB gold standard variant calls. We also demonstrate that EMA’s alignments improve phasing performance over Lariat’s in both NA12878 and NA24385, producing fewer switch/mismatch errors and larger phased blocks on average.

Moreover, we demonstrate that EMA is able to effectively resolve alignments in regions containing nearby homologous elements—a particularly challenging problem in read mapping—through the introduction of our novel statistical binning optimization framework, which enables us to find variants in the pharmacogenomically important CYP2D region that go undetected when using Lariat or BWA. This enhanced capability addresses one of the major weaknesses of linked-read sequencing as compared to long-read sequencing, where only a relatively small subset of the original source fragment is observed—and more specifically, that the order of reads within the fragment is not known—making it difficult to produce accurate alignments if the fragment spans homologous elements.

Discussion: Our advance is a general framework applicable to many barcoded sequencing problems. It is likely to be of interest to any developers, and even users, of barcoded or linked-read sequencing technologies that come along. We highlight that 10x sequencing is just an instance of general “barcoded read sequencing”, and other technologies that make use of the same paradigm already exist and are likely to emerge in the future, given its numerous advantages over long-read sequencing. Several technologies already employ barcoded sequencing in addition to 10x Genomics’, such as Illumina’s TruSeq SLR platform (formerly Moleculo), and Complete Genomics’ Long Fragment technology. Our framework should apply to these (and similar) technologies as well. Due to their substantial improvements over existing methods for aligning and interpreting linked-read data, the algorithms employed by EMA are likely to be a fundamental component of read cloud-based methods in the future.

Acknowledgements. We thank Chris Whelan, Chad Nusbaum, Eric Banks, as well as the rest of the SV Group from the Broad Institute for providing us with data samples and many valuable suggestions. Also, we thank Jian Peng and Lillian Zhang for their helpful suggestions.

Funding A.S., I.N. and B.B. are partially funded by NIH grant GM108348.

References

1. Bishara, A., et al.: Read clouds uncover variation in complex regions of the human genome. *Genome Res* **25**(10), 1570–1580 (2015)
2. Sekar, A., et al.: Schizophrenia risk from complex variation of complement component 4. *Nature* **530**, 177 (2016)



ModulOmics: Integrating Multi-Omics Data to Identify Cancer Driver Modules

Dana Silverbush¹(✉), Simona Cristea^{2,3,4}(✉), Gali Yanovich⁵, Tamar Geiger⁵,
Niko Beerenwinkel^{6,7}, and Roded Sharan¹

¹ Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel
dsilverb@broadinstitute.org, roded@post.tau.ac.il

² Department of Biostatistics and Computational Biology,
Dana-Farber Cancer Institute, Boston, MA, USA

³ Department of Biostatistics, Harvard T.H. Chan School of Public Health,
Boston, MA, USA

⁴ Department of Stem Cell and Regenerative Biology, Harvard University,
Cambridge, MA, USA
scristea@jimmy.harvard.edu

⁵ Department of Human Molecular Genetics and Biochemistry,
Sackler Faculty of Medicine, Tel Aviv University, Tel Aviv, Israel

⁶ Department of Biosystems Science and Engineering, ETH Zurich, Basel,
Switzerland

niko.beerenwinkel@bsse.ethz.ch

⁷ Swiss Institute of Bioinformatics, Basel, Switzerland

Introduction: Recent rapid advancements in sequencing technologies allowed the collection of DNA, RNA, and protein data from tens of thousands of cancer patients. Mathematical and computational tools are used to analyze these complex data sets, aiming to reveal mechanistic and predictive insights into tumor treatment and progression. Key to achieving these goals is finding molecular alterations that drive tumorigenesis, or drivers, such as single nucleotide variants (SNVs), copy number alterations (CNAs), changes in the transcriptional activity of genes, or changes in protein concentration. Groups of such functionally connected genetic alterations, also termed cancer driver modules or pathways, activate mechanisms that gradually contribute to triggering the hallmarks of cancer, conferring fitness advantages to the tumors. The identification of such driver modules is an important challenge in the field of cancer genomics, since clinically targeting driver pathways can improve patient treatment. Nevertheless, most of the existing computational tools to address this problem use primarily somatic mutations, not fully exploiting additional data types. Here, we describe ModulOmics, a method to *de novo* identify cancer driver modules by integrating multiple sources of biological information (protein-protein interactions, mutual exclusivity of mutations or copy number alterations, transcriptional co-regulation, and RNA co-expression) into a single probabilistic model.

Methods: Given a set $G = \{G_1, \dots, G_n\}$ of genes and a collection $M = \{M_1, \dots, M_m\}$ of models for different data types, we introduce S_G ,

D. Silverbush, S. Cristea, N. Beerenwinkel, and R. Sharan — equal contribution.

the ModulOmics probabilistic score of the set G , reflecting how likely are the genes in G to be functionally connected. S_G is computed as the mean of m probabilistic scores $P(G | M_k)$, each representing the degree of functional connectivity of the set G , under a different model:

$$S_G = \frac{1}{m} \sum_{k=1}^m P(G | M_k) \quad (1)$$

Here, we consider four models, as follows: M_1 computes the connectivity of the genes in G based on their proximity in the protein-protein interaction (PPI) network, M_2 estimates the degree of mutual exclusivity among DNA alterations of the genes in G across the patient cohort, M_3 assesses the co-regulation of the genes in G on the basis of their shared transcriptional regulators that are active in the patient cohort, and M_4 evaluates the transcriptional connectivity of the genes in G based on their coexpression profiles. The goal of ModulOmics is to identify groups that maximize the global score in Eq. 1. As the number of candidate groups grows exponentially with maximal group size, we use a heuristic two-step optimization procedure. The optimization routine first performs an approximation of the exact scores of the set G under each of the four models M_k , by decomposing them into pairwise scores and using integer linear programming (ILP) to find good initial solutions. The initial solutions are further refined via stochastic search starting from these initial solutions and using the global score.

Results: Using ModulOmics, we accurately identify known cancer driver genes and pathways in three large-scale TCGA datasets of breast cancer, glioblastoma (GBM) and ovarian cancer, outperforming state-of-the-art methods for module detection. Notably, in breast cancer subtypes, the highest scoring modules reliably separate cancerous from normal tissues in an independent patient cohort. Focusing on individual subtypes, the modules of Her2 and Basal are enriched with Gene Ontology (GO) terms related to cell proliferation, reflecting their more aggressive nature. Driver modules in triple negative (TN) samples capture the accumulation of down-regulated tumor suppressors such as *TP53*, *BRCA1*, *RB1* and *PTEN*, a pattern also supported by reverse phase protein array (RPPA) data. The highest scoring modules in Luminal A suggest two potential functionalities of *PTEN*: a canonical one as part of the PI3K pathway, and a non-canonical one as a regulator of cell proliferation. ModulOmics is freely available in two forms, as an open-source R code for the identification of cancer driver modules from a cohort of cancer samples (<https://github.com/danasily/ModulOmics>), and as a webserver for the evaluation of any set of genes of interest using the TCGA data processed in this study (<http://anat.cs.tau.ac.il/ModulOmicsServer/>).



SCI Φ : Single-Cell Mutation Identification via Phylogenetic Inference

Jochen Singer^{1,2}, Jack Kuipers^{1,2}, Katharina Jahn^{1,2},
and Niko Beerenwinkel^{1,2}(✉)

¹ Department of Biosystems Science and Engineering,
ETH Zurich, Basel, Switzerland
niko.beerenwinkel@bsse.ethz.ch

² SIB Swiss Institute of Bioinformatics, Basel, Switzerland

Abstract. Understanding the evolution of cancer is important for the development of appropriate cancer therapies. The task is challenging because tumors evolve as heterogeneous cell populations with an unknown number of genetically distinct subclones of varying frequencies. Conventional approaches based on bulk sequencing are limited in addressing this challenge as clones cannot be observed directly. Single-cell sequencing holds the promise of resolving the heterogeneity of tumors. However, this advantage comes at the cost of elevated noise due to the limited amount of DNA material present in a cell and the extensive DNA amplification required prior to sequencing.

Here, we present SCI Φ , the first single-cell-specific variant caller that combines single-cell genotyping with reconstruction of the cell lineage tree. SCI Φ leverages the fact that the somatic cells of an organism are related via a phylogenetic tree where mutations are propagated along tree branches. Our inference scheme starts with an initial identification of possible mutation loci and then performs joint phylogenetic inference and variant calling via posterior sampling.

In a first step, likely mutated loci are identified using the posterior probability of observing at least one mutated cell at a specific locus. In order to do so, SCI Φ models the nucleotide counts using a beta-binomial distribution. This is especially useful in the single-cell setting, since the beta-binomial distribution can be described as a Pólya urn model, which in turn is a very close approximation of the multiple displacement amplification commonly used to amplify the genomic material of a single-cell.

In a second step, the identified loci are used to infer the tumor phylogeny. Here, we account for dropout events by modeling the likelihood of observing a mutation in a cell as a weighted mixture of the likelihoods of homozygous reference genotype, heterozygous genotype, and homozygous alternative genotype. Our model to infer tumor phylogeny consists of three parts: the genealogical tree, the mutation attachments to edges, and the parameters of the model. Because the tree search space grows superexponentially in the number of cells, we employ a Markov Chain

J. Singer, J. Kuipers — These authors contributed equally.

Monte Carlo scheme to traverse through the tree space with mutation assignment and learn the parameters of the model.

Using the relationship between cells, we are able to reliably call mutations in each single-cell even in experiments with high dropout rates and missing data. We show that SCI Φ outperforms existing methods on simulated data and apply it to different real-world datasets. Availability: <https://github.com/cbg-ethz/SCIPhi>



AptaBlocks: Accelerating the Design of RNA-Based Drug Delivery Systems

Yijie Wang¹, Jan Hoinka¹, Piotr Swiderski², and Teresa M. Przytycka¹(✉)

¹ National Center of Biotechnology Information,
National Library of Medicine, NIH, Bethesda, MD 20894, USA
przytyck@ncbi.nlm.nih.gov

² Department of Molecular and Cellular Biology,
Beckman Research Institute of City of Hope, Duarte, CA 91010, USA

Extended Abstract

Synthetic RNA molecules are increasingly used to alter cellular functions [1–4]. These successful applications indicate that RNA-based therapeutics might be able to target currently undruggable genes [5, 6]. However, to achieve this promise, an effective method for delivering therapeutic RNAs into specific cells is required. Recently, RNA aptamers emerged as promising delivery agents due to their ability of binding specific cell receptors [7, 8]. Crucially, these aptamers can frequently be internalized into the cells expressing these receptors on their surfaces. This property is leveraged in aptamer based drug delivery systems by combining such receptor-specific aptamers with a therapeutic “cargo” such that the aptamer facilitates the internalization of the cargo into the cell [9–11]. The advancement of this technology however is contingent on an efficient method to produce stable molecular complexes that include specific aptamers and cargoes. A recently proposed experimental procedure for obtaining such complexes relies on conjugating the aptamer and the cargo with complementary RNA strands so that when such modified molecules are incubated together, the complementary RNA strands hybridize to form a double-stranded “sticky bridge” connecting the aptamer with its cargo [12, 13]. However, designing appropriate sticky bridge sequences guaranteeing the formation and stability of the complex while simultaneously not interfering with the aptamer or the cargo as well as not causing spurious aggregation of the molecules during incubation has proven highly challenging.

To fill this gap, we developed AptaBlocks, a computational method to design sticky bridges to connect RNA-based molecules (blocks). Accounting for the three-step procedure [12, 13], we formulate the sticky bridge sequence design as an optimization problem utilizing an objective function which reflects the biophysical characteristics of the assembly process. Specifically, we designed the objective function considering the equilibrium probabilities of the target structures over all possible structures of the aptamer-stick and cargo-stick, the probability of the interaction between the aptamer-stick and cargo-stick at equilibrium, the hybridization energy between the sticky bridge sequences, and additional

sequence constraints including but not limited to the GC content. We further provide a simulated annealing algorithm that enables efficient estimation of the corresponding combinatorial optimization problem. The effectiveness of the algorithm has been verified computationally and experimentally. AptaBlocks can be used in a variety of experimental settings and its preliminary version has already been leveraged to design an aptamer based delivery system for a cytotoxic drug targeting Pancreatic ductal adenocarcinoma cells [14]. It is thus expected that AptaBlocks will play a substantial role in accelerating RNA-based drug delivery design.

References

1. Kushwaha, M., et al.: Using RNA as molecular code for programming cellular function. *ACS Synth. Biol.* **5**(8), 795–809 (2016)
2. Chappell, J., Watters, K.E., Takahashi, M.K.: A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future. *Curr. Opin. Chem. Biol.* **28**, 47–56 (2015)
3. Mckeague, M., Wong, R.S., Smolke, C.D.: Opportunities in the design and application of RNA for gene expression control. *Nucleic Acids Res.* **44**(10), 2987–2999 (2016)
4. Qi, L.S., Arkin, A.P.: A versatile framework for microbial engineering using synthetic non- coding RNAs. *Nat. Rev. Microbiol.* **12**(5), 341–354 (2014)
5. Ryther, R.C.C., et al.: siRNA therapeutics: big potential from small RNAs. *Gene Ther.* **17**(1), 5–11 (2005)
6. Chakraborty, C.: Potentiality of small interfering RNAs (siRNA) as recent therapeutic targets for. *Curr. Drug Targets* **8**(3), 469–482 (2007)
7. Zhou, J., Rossi, J.J.: Cell-specific aptamer-mediated targeted drug delivery. *Oligonucleotides* **21**(1), 1–10 (2011)
8. Zhang, Y., Hong, H., Cai, W.: Tumor-targeted drug delivery with aptamers. *Curr. Med. Chem.* **18**(27), 4185–4194 (2011)
9. Mcnamara II, J.O., et al.: Cell type-specific delivery of siRNAs with aptamer siRNA chimeras. *Nat. Biotechnol.* **24**(8), 1005–1015 (2006)
10. Thiel, K.W., et al.: Delivery of chemo-sensitizing siRNAs to HER2 + -breast cancer cells using RNA aptamers. *Nucleic Acids Res.* **40**(13), 6319–6337 (2012)
11. Pastor, F., et al.: Induction of tumour immunity by targeted inhibition of nonsense-mediated mRNA decay. *Nature* **465**(7295), 227–230 (2010)
12. Zhou, J., et al.: Selection, characterization and application of new RNA HIV gp 120 aptamers for facile delivery of Dicer substrate siRNAs into HIV infected cells. *Nucleic Acids Res.* **37**(9), 3094–3109 (2009)
13. Zhou, J., Rossi, J.: Aptamers as targeted therapeutics: current potential and challenges. *Nat. Rev. Drug Discov.* **16**(3), 181–202 (2016)
14. Yoon, S., et al.: Aptamer-drug conjugates of active metabolites of nucleoside analogs and cytotoxic agents inhibit pancreatic tumor cell growth. *Mol. Ther.: Nucleic Acid* **6**, 80–88 (2017)



A Unifying Framework for Summary Statistic Imputation

Yue Wu¹, Eleazar Eskin^{1,2}, and Sriram Sankararaman^{1,2}(✉)

¹ Department of Computer Science, UCLA, Los Angeles, USA
{eeskin,sriram}@cs.ucla.edu

² Department of Human Genetics, UCLA, Los Angeles, USA

Imputation has been widely utilized to aid and interpret the results of Genome-Wide Association Studies (GWAS). Imputation methods, that aim to fill in “data” at untyped SNPs, have emerged as an effective strategy to increase the power of GWAS since the causal variant may not be directly observed or typed in these studies. In the context of GWAS, there are two broad classes of methods to impute association statistics at untyped SNPs. The first class, termed **Two-step imputation**, imputes genotypes at untyped SNPs followed by computing association statistics at the imputed genotypes [1–6]. In practice, the first step of genotype imputation relies on discrete Hidden Markov Models (HMM) [1, 6]. The second class of methods, termed *summary statistic imputation (SSI)*, directly imputes association statistics at untyped SNPs given the association statistics at the typed SNPs. The joint distribution of association statistics at the typed SNPs and untyped SNPs has been shown to follow a multivariate normal distribution (MVN) [7–9]. **SSI** is appealing as it tends to be computationally efficient while only requiring the summary statistics from a study while the **Two-step imputation** methods require access to individual-level data which can be difficult to obtain in practice.

Current summary-statistic based imputation methods calibrate the imputed statistics using a technique we call *variance re-weighting (SSI-VR)*. Despite recent progress, the statistical properties of summary statistic imputation methods (including the impact of variance re-weighting) and the connection between the two classes of summary statistic imputation methods has not been adequately understood.

In this paper, we show that the two classes of imputation methods, **Two-step imputation** and **SSI** are asymptotically multivariate normal with small differences in the underlying covariance matrix. Using this asymptotic equivalence, we can understand the effect of the imputation method on the power of the study. Our new method, **SSI**, performs summary statistic imputation without variance re-weighting. The resulting statistics do not then have unit variance as in traditional summary statistic imputation but instead correctly take into account the ambiguity of the imputation process.

We compared the performance of the different imputations methods on the Northern Finland Birth Cohort (NFBC) data set [10] to show that **SSI** increases power over no imputation while SSI-VR can sometimes lead to lower power.

Finally, we compared the results from **SSI**, **SSI-VR** and **Two-step imputation** on the NFBC dataset and show that the resulting statistics are close thereby justifying the theory.

References

1. Browning, S.R., Browning, B.L.: Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering. *Am. J. Hum. Genet.* **81**, 1084–1097 (2007)
2. Howie, B., Fuchsberger, C., Stephens, M., Marchini, J., Abecasis, G.R.: Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nat. Genet.* **44**(8), 955–959 (2012)
3. 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)
4. Li, Y., Willer, C., Sanna, S., Abecasis, G.: Genotype imputation. *Annu. Rev. Genomics Hum. Genet.* **10**, 387–406 (2009)
5. Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R.: MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet. Epidemiol.* **34**(8), 816–834 (2010)
6. Marchini, J., Howie, B., Myers, S., McVean, G., Donnelly, P.: A new multipoint method for genome-wide association studies by imputation of genotypes. *Nat. Genet.* **39**, 906–913 (2007)
7. Han, B., Kang, H.M., Eskin, E.: Rapid and accurate multiple testing correction and power estimation for millions of correlated markers. *PLoS Genet.* **5**(4), e1000456 (2009)
8. Kostem, E., Lozano, J.A., Eskin, E.: Increasing power of genome-wide association studies by collecting additional single-nucleotide polymorphisms. *Genetics* **188**(2), 449–460 (2011)
9. Hormozdiari, F., Kostem, E., Kang, E.Y., Pasaniuc, B., Eskin, E.: Identifying causal variants at loci with multiple signals of association. *Genetics* **198**(2), 497–508 (2014)
10. Sabatti, C., Hartikainen, A.-L., Pouta, A., et al.: Genome-wide association analysis of metabolic traits in a birth cohort from a founder population. *Nat. Genet.* **41**(1), 35–46 (2009)



Characterizing Protein-DNA Binding Event Subtypes in ChIP-Exo Data

Naomi Yamada, William K. M. Lai, Nina Farrell, B. Franklin Pugh,
and Shaun Mahony^(✉)

Department of Biochemistry and Molecular Biology, Center for Eukaryotic Gene Regulation, The Pennsylvania State University, University Park, PA 16802, USA
mahony@psu.edu

Introduction: A given regulatory protein may have multiple modes of interaction with the genome; at some sites, it may directly bind cognate DNA motifs, while at others it may bind indirectly via protein-protein interactions with other regulators. Each protein-DNA interaction mode may be associated with distinct sequence motifs, and may also produce distinct patterns in high-resolution protein-DNA binding assays. For example, the ChIP-exo [1] protocol precisely characterizes protein-DNA crosslinking patterns by combining chromatin immunoprecipitation (ChIP) with 5' to 3' exonuclease digestion. Since different regulatory complexes will result in different protein-DNA crosslinking signatures, analysis of ChIP-exo sequencing tag patterns should enable detection of multiple protein-DNA binding modes for a given regulatory protein. However, current ChIP-exo analysis methods either treat all binding events as being of a uniform type, or rely on DNA motifs to cluster binding events into subtypes.

We introduce the ChIP-exo mixture model (ChExMix) to systematically detect multiple protein-DNA interaction modes in a single ChIP-exo experiment. ChExMix discovers and characterizes binding event subtypes in ChIP-exo data by leveraging both sequencing tag enrichment patterns and DNA motifs. ChExMix defines possible binding event subtypes by both clustering observed ChIP-exo tag distribution patterns and performing targeted *de novo* motif discovery around the positions of the predicted binding events. ChExMix then uses an Expectation Maximization learning scheme to probabilistically model the genomic locations and subtype membership of binding events using both ChIP-exo tag locations and DNA sequence information. In analyzing ChIP-exo data, ChExMix offers a more principled and robust approach to characterizing binding subtypes than simply clustering binding events using motifs.

Results: ChExMix uses DNA motif and ChIP-exo tag distribution patterns to accurately estimate multiple binding subtypes within a single ChIP-exo. We demonstrate the ability of ChExMix to estimate binding subtypes and assign binding events to subtypes by creating datasets that computationally mix data from CTCF and FoxA1 ChIP-exo experiments. CTCF and FoxA1 are known to display distinct ChIP-exo tag distribution patterns at their respective binding events. We simulated different representations of each subtype by modulating the relative number of tags drawn from each ChIP-exo experiment. ChExMix detects the two subtypes and accurately assigns subtypes to binding events over a wide range of relative sampling rates from the CTCF and FoxA1 subtypes. In contrast, a motif-driven approach fails to appropriately classify

many of the FoxA1 subtype binding events. ChExMix performance remains reasonably high when we remove DNA motifs from consideration and assign subtypes using only ChIP-exo tag distribution information. Our results demonstrate that ChExMix enables discovery of unique subtypes within a single ChIP-exo dataset and accurately assigns subtypes to binding events.

To assess ChExMix's ability to characterize binding locations, we compare ChExMix performance in predicting human CTCF and mouse FoxA2 binding event locations to that of seven ChIP-exo analysis methods. ChExMix outperforms other methods by exactly locating the CTCF events at the motif position in 90.2% of the shared CTCF events. Similarly, ChExMix exactly locates the FoxA2 events at the motif position in 67.4% of the shared FoxA2 events. ChExMix binding event predictions also contain instances of the cognate motif at a high rate. These results suggest that ChExMix maintains high accuracy in protein-DNA binding event predictions.

We further demonstrate that ChExMix can characterize biologically relevant binding event subtypes in ER positive breast cancer cells. FoxA1, ER α , and CTCF have previously been shown to co-localize at a subset of genomic loci. However, how these proteins interact with each other and DNA at specific sites remained elusive. In FoxA1 ChIP-exo data, ChExMix identifies subtypes corresponding to ER α and CTCF motifs, and about a half of these subtypes' binding events display ER α and CTCF ChIP-exo enrichment with similar tag distributions. Our results thus suggest that ER α and CTCF may mediate binding of FoxA1 via protein-protein interactions at a subset of the genomic loci where multiple factors are co-bound. These results strongly suggest that ChExMix can discover binding event subtypes representing direct and indirect TF interactions from a single ChIP-exo experiment.

Conclusions: ChExMix provides a principled platform for elucidating diverse protein-DNA interaction modes in a single ChIP-exo experiment by exploiting both ChIP-exo tag enrichment patterns and DNA motifs. Using a fully integrated framework, ChExMix allows simultaneous detection of binding event locations, discovery of binding event subtypes, and assignment of binding events to subtypes. ChExMix enables new forms of insight from a single ChIP-exo experiment, taking analysis towards a fine-grained characterization of distinct protein-DNA binding modes at specific genomic loci. ChExMix is freely available from <https://github.com/seqcode/chexmix>.

Reference

1. Rhee, H.S., Pugh, B.F.: Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. *Cell* **147**(6), 1408–1419 (2011)



Continuous-Trait Probabilistic Model for Comparing Multi-species Functional Genomic Data

Yang Yang¹, Quanquan Gu², Takayo Sasaki³, Julianna Crivello⁴,
Rachel O'Neill⁴, David M. Gilbert³, and Jian Ma¹(✉)

¹ Computational Biology Department, School of Computer Science,
Carnegie Mellon University, Pittsburgh, USA
jianma@cs.cmu.edu

² Department of Computer Science, University of Virginia, Charlottesville, USA

³ Department of Biological Science, Florida State University, Tallahassee, USA

⁴ Department of Molecular and Cell Biology, Institute for Systems Genomics,
University of Connecticut, Storrs, USA

Multi-species functional genomic data from various high-throughput assays are highly informative for the comparative analysis of gene regulation to better understand the molecular mechanisms of phenotypic diversity between human and other mammalian species. Continuous-trait models, which are key to the modeling of functional genomic signals, are gaining increasing attention in genome-wide comparative genomic studies. However, computational models are currently under-explored to fully capture continuous features in the context of multi-species comparisons. There have been several types of continuous-trait evolutionary models, including Brownian motion and Ornstein-Uhlenbeck (OU) process. However, to the best of our knowledge, there are no existing computational methods available to simultaneously infer heterogeneous continuous-trait evolutionary models along the genome based on functional genomic signals.

In this paper, we develop a new continuous-trait probabilistic model for more accurate state estimation using multi-variate features from cross-species functional genomic signals. We call our model phylogenetic hidden Markov Gaussian processes (Phylo-HMGP). Phylo-HMGP incorporates the evolutionary affinity among multiple species into the hidden Markov model (HMM) for exploiting both temporal dependencies across species in the context of evolution and spatial dependencies along the genome in a continuous-trait model. The goal of the proposed method is to identify heterogeneous cross-species genomic feature patterns more effectively. The Gaussian processes embedded in the HMM are specialized to be multi-variate OU processes or Brownian motion in this study.

Both simulation studies and real data application demonstrate the effectiveness of Phylo-HMGP. Importantly, we applied Phylo-HMGP to analyze a new cross-species DNA replication timing (RT) dataset from the same cell type in five primate species (human, chimpanzee, orangutan, gibbon, and green monkey). We demonstrate that our Phylo-HMGP model enables discovery of genomic regions with distinct evolutionary patterns of RT. We found that regions with

conserved early RT and conserved late RT exhibit strong correlation with constitutive early RT and constitutive late RT, respectively, defined from human ES cell differentiation. In addition, we found enrichment for specific *cis*-regulatory elements in hominini specific early RT regions.

Taken together, the proposed Phylo-HMGP explores a new integrative framework to utilize continuous-trait evolutionary models with spatial constraints to study genome-wide functional genomic features across species. The new method is also flexible such that varied continuous-trait evolutionary models or assumptions can be incorporated. We believe that Phylo-HMGP provides a generic framework that has the potential to more precisely capture the evolutionary history of regulatory regions based on functional genomic signals across different species.



Deep Learning Reveals Many More Inter-protein Residue-Residue Contacts than Direct Coupling Analysis

Tian-Ming Zhou^{1,2}, Sheng Wang³(✉), and Jinbo Xu¹(✉)

¹ Toyota Technological Institute at Chicago, Chicago, USA
jinboxu@gmail.com

² The Institute for Theoretical Computer Science (ITCS), Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

³ Computational Bioscience Research Center (CBRC), King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

We study how to predict inter-protein residue-residue contacts between a pair of putative interacting proteins, which has been reported useful for the 3D structure modeling of a PPI or protein docking. Direct-coupling analysis (DCA) has been applied to intra-protein and inter-protein contact prediction, but it does not fare well for proteins without many sequence homologs. This is a big issue for inter-protein contact prediction since it is challenging to find so many interlogs (i.e., interacting homologs). Because of this, currently DCA for inter-protein contact prediction mainly focuses on prokaryotes and mitochondria [1, 2] since it is relatively easy to find interlogs in prokaryotes, but not in eukaryotes with abundant paralogs.

We have developed a deep learning (DL) method for intra-protein contact prediction [3–5], which greatly outperformed DCA and was officially ranked first in CASP12 [6]. Our DL method needs much fewer sequence homologs than DCA to be effective because it makes use of contact occurrence patterns, in addition to co-evolution, for contact prediction. This abstract shows that DL can also work on inter-protein contact prediction, especially for eukaryotes. To avoid overfitting, we do not train our DL model using any protein complex data (i.e., inter-protein contacts), but use our previous DL model trained by only protein chains (i.e., intra-protein contacts) to predict inter-protein contacts.

We propose a new phylogeny-based method to identify interlogs for a putative interacting protein pair, especially for eukaryotes in which some interacting genes may have big genomic distance. Coupled with DL, this new method works better on eukaryotes than genome-based methods employed by Baker [1] and Marks [2].

As shown in Fig. 1, given a pair of putative interacting proteins A and B under prediction, we first build multiple sequence alignments (MSAs) for A and B, respectively. Then we employ genome- and phylogeny-based strategies to concatenate MSA_A and MSA_B into two paired MSAs consisting of only interlogs. Finally, we use our DL method to predict two inter-protein contact maps and average them for final prediction. Our DL method outperforms pure DCA on three large datasets and works on both prokaryotes and eukaryotes. Table 1 shows the performance comparison on Baker's dataset.

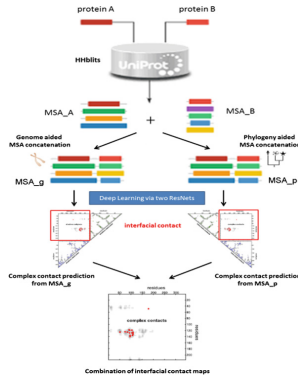


Fig. 1. Method flowchart

Table 1. Inter-protein contact prediction accuracy (%) on Baker’s data. GCNN is our method and (s) indicates a web server. EVfold is same as EVcomplex, but run locally with our MSAs. “Genome” and “Phylogeny” denote two MSA generation methods. “Merged” indicates prediction is merged from “Genome” and “Phylogeny”. Columns 3–9 show accuracy of top L/10, L/20, 20 and 10 predicted contacts.

Predictor	MSA	L/10	L/20	20	10
EVcomplex(s)	Built-in	14.25	20.10	21.55	26.55
Gremlin(s)	Built-in	23.74	33.23	41.21	52.76
EVfold	Genome	28.01	39.45	46.90	57.59
EVfold	Phylogeny	15.61	23.09	26.21	36.21
EVfold	Merged	25.13	36.12	42.07	54.83
CCMpred	Genome	28.44	39.54	47.41	53.45
CCMpred	Phylogeny	17.04	25.49	30.34	39.31
CCMpred	Merged	27.70	38.72	46.03	55.52
GCNN	Genome	51.41	60.80	62.76	68.79
GCNN	Phylogeny	32.61	39.30	42.24	47.59
GCNN	Merged	48.25	57.09	60.52	65.86

References

1. Ovchinnikov, S., Kamisetty, H., Baker, D.: Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *Elife* **3**, e02030 (2014)
2. Hopf, T.A., et al.: Sequence co-evolution gives 3D contacts and structures of protein complexes. *Elife* **3**, e03430 (2014)
3. Wang, S., et al.: Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput. Biol.* **13**(1), e1005324 (2017)
4. Wang, S., et al.: Folding membrane proteins by deep transfer learning. *Cell Syst.* **5**(3), 202–211. e3 (2017)
5. Wang, S., Sun, S., Xu, J.: Analysis of deep learning methods for blind protein contact prediction in CASP12. *Proteins: Struct. Funct. Bioinf.* (2017)
6. Schaarschmidt, J., et al.: Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age. *Proteins* (2017)

Author Index

- Achtman, Mark 225
Alikhan, Nabil-Fareed 225
Almodaresi, Fatemeh 271
Altenbuchinger, Michael 75
Aluru, Chaitanya 211
Aluru, Srinivas 211
- Bakhtiari, Mehrdad 243
Bankevich, Anton 1
Bafna, Vineet 243, 276
Bansal, Vikas 243
Basso, Rebecca Sarto 278
Beltran, Pierre M. Jean 54
Beerenwinkel, Niko 269, 283, 285
Berger, Bonnie 245, 251, 280, 285
Bender, Michael A. 271
Bepler, Tristan 245
Bohmann, Kristine 276
Bonnet, Édouard 248
Borojeny, Ali Ebrahimpour 37
Brasch, Julia 245
- Canzar, Stefan 21
Cannistra, Anthony 263
Chakraborty, Shounak 21
Chikhi, Rayan 105
Cho, Hyunghoon 251
Chitsaz, Hamidreza 37
Chockalingam, Sriram P. 211
Craven, Mark 194
Cristea, Ileana M. 54
Cristea, Simona 283
Crivello, Julianna 293
Crovella, Mark 263
- DeCourcy, Alex 138
Durif, G. 254
- Eskin, Eleazar 289
- Fan, Jason 263
Farrell, Nina 291
Ferdman, Michael 271
- Franklin Pugh, B. 291
Fried, Inbar 263
- Gagie, Travis 105
Galitzine, Cyril 54
Gallagher, Suzanne Renick 37
Gasch, Audrey 194
Geiger, Tamar 283
Gilbert, David M. 293
Gilbert, M. Thomas P. 276
Görtler, Franziska 75
Gu, Quanquan 293
Gymrek, Melissa 243
- Halperin, Eran 274
Hammer, Stefan 256
Hescott, Benjamin 263
Ho, Yi-Hsuan 194
Hochbaum, Dorit S. 278
Hoinka, Jan 287
Hormozdiari, Fereydoun 259
Huynh, Linh 259
- Ideker, Trey 266
- Jahn, Katharina 269, 285
Johnson, Rob 271
Joseph, Tyler A. 90
- Kolmogorov, Mikhail 261
Kuipers, Jack 269, 285
Kuosmanen, Anna 105
- Lai, William K. M. 291
Lambert-Lacroix, S. 254
Larson, Gary 122
Leiserson, Mark D. M. 263
Li, Sujun 138
Lim, Tim 263
Lin, Yu 261
Liu, Yang 266
Luhmann, Nina 225
Luo, Yunan 266

- Ma, Jian 293
 Ma, Jianzhu 266
 Mahony, Shaun 291
 Mäkinen, Veli 105
 Malikic, Salem 269
 Marschall, Tobias 21
 Mirarab, Siavash 276
 Modolo, L. 254
 Mold, J. E. 254
 Morin, Andrew 245
- Noble, Alex J. 245
 Numanagić, Ibrahim 280
- Oefner, Peter J. 75
 O'Neill, Rachel 293
 Orenstein, Yaron 154
- Pandey, Prashant 271
 Patro, Rob 271
 Paavilainen, Topi 105
 Pe'er, Itsik 90
 Peng, Jian 251, 266
 Pevzner, Pavel 1, 261
 Picard, F. 254
 Ponty, Yann 256
 Przytycka, Teresa M. 287
- Quince, Christopher 225
- Rahmani, Elior 274
 Roch, Sebastien 167
 Rosset, Saharon 274
 Rzażewski, Paweł 248
- Sankararaman, Sriram 274, 289
 Sarmashghi, Shahab 276
 Sahinalp, S. Cenk 37, 269
 Sasaki, Takayo 293
 Schaffner, Thomas 263
 Schmidler, Scott 122
 Schulz, Marcel H. 21
 Schweiger, Regev 274
 Shajii, Ariya 280
- Shapiro, Lawrence 245
 Sharan, Roded 283
 Sharifi-Zarchi, Ali 37
 Shleizer-Burko, Sharona 243
 Shrestha, Akash 37
 Sikora, Florian 248
 Singer, Jochen 285
 Silverbush, Dana 283
 Solbrig, Stefan 75
 Soulé, Antoine 177
 Spang, Rainer 75
 Steyaert, Jean-Marc 177
 Sverchkov, Yuriy 194
 Swiderski, Piotr 287
- Tang, Haixu 138
 Thankachan, Sharma V. 211
 Thorne, Jeffrey L. 122
 Tomescu, Alexandru 105
- Vandin, Fabio 278
 Vitek, Olga 54
- Waldispühl, Jérôme 177
 Wang, Kun-Chieh 167
 Wang, Sheng 295
 Wang, Wei 256
 Wang, Yijie 287
 Wettig, Tilo 75
 Will, Sebastian 256
 Wu, Yue 289
- Xu, Jinbo 295
- Yamada, Naomi 291
 Yang, Yang 293
 Yanovich, Gali 283
 Ye, Qing 266
 Yuan, Jeffrey 261
- Zhou, Tian-Ming 295
 Zhou, Zhemín 225