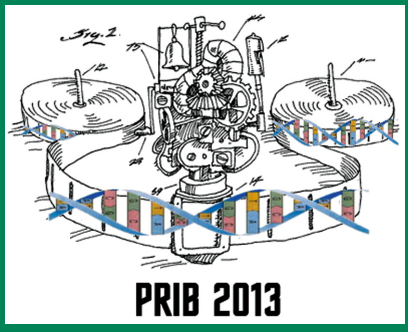
Alioune Ngom
Enrico Formenti
Jin-Kao Hao
Xing-Ming Zhao
Twan van Laarhoven (Eds.)

LNBI 7986

Pattern Recognition in Bioinformatics

8th IAPR International Conference, PRIB 2013
Nice, France, June 2013
Proceedings



Springer

Lecture Notes in Bioinformatics 7986
Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Alioune Ngom Enrico Formenti
Jin-Kao Hao Xing-Ming Zhao
Twan van Laarhoven (Eds.)

Pattern Recognition in Bioinformatics

8th IAPR International Conference, PRIB 2013
Nice, France, June 17-20, 2013
Proceedings

Springer

Volume Editors

Alioune Ngom
University of Windsor, ON, Canada
E-mail: angom@uwindsor.ca

Enrico Formenti
Nice Sophia Antipolis University, France
E-mail: enrico.formenti@unice.fr

Jin-Kao Hao
Université d'Angers, France
E-mail: hao@info.univ-angers.fr

Xing-Ming Zhao
Tongji University, Shanghai, China
E-mail: zhaoxingming@gmail.com

Twan van Laarhoven
Radboud University, Nijmegen, The Netherlands
E-mail: tvanlaarhoven@cs.ru.nl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-39158-3 e-ISBN 978-3-642-39159-0
DOI 10.1007/978-3-642-39159-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013941044

CR Subject Classification (1998): J.3, I.5, F.2.2, I.2, I.4, H.3.3, H.2.8

LNCS Sublibrary: SL 8 – Bioinformatics

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

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

In the post-genomic era, a holistic understanding of biological systems and processes, in all their complexity, is critical in comprehending nature's choreography of life. As a result, bioinformatics involving its two main disciplines, namely, the life sciences and the computational sciences, is fast becoming a very promising multidisciplinary research field. With the ever-increasing application of large-scale high-throughput technologies, such as gene or protein microarrays and mass spectrometry methods, the enormous body of information is growing rapidly. Bioinformaticians are posed with a large number of difficult problems to solve, arising not only due to the complexities in acquiring the molecular information but also due to the size and nature of the generated data sets and/or the limitations of the algorithms required for analyzing these data. The recent advancements in computational and information-theoretic techniques are enabling us to conduct various in silico testing and screening of many lab-based experiments before these are actually performed in vitro or in vivo. These in silico investigations are providing new insights for interpreting and establishing new direction for a deeper understanding. Among the various advanced computational methods currently being applied to such studies, the *pattern recognition* techniques are mostly found to be at the core of the whole discovery process for apprehending the underlying biological knowledge. Thus, we can safely surmise that the ongoing bioinformatics *revolution* may, in future, inevitably play a major role in many aspects of medical practice and/or the discipline of life sciences.

The aim of this conference on Pattern Recognition in Bioinformatics (PRIB) is to provide an opportunity to academics, researchers, scientists, and industry professionals to present their latest research in pattern recognition and computational intelligence-based techniques applied to problems in bioinformatics and computational biology. It also provides them with an excellent forum to interact with each other and share experiences. The conference is organized jointly by the Nice Sophia Antipolis University, France, and IAPR (International Association for Pattern Recognition) Bioinformatics Technical Committee (TC-20).

This volume presents the proceedings of the 8th IAPR International Conference on Pattern Recognition in Bioinformatics (PRIB 2013), held in Nice, June 17–19, 2013. It includes 25 technical contributions that were selected by the International Program Committee from 43 submissions. Each of these rigorously reviewed papers was presented orally at PRIB 2013. The proceedings consists of five parts:

Part I Bio-Molecular Networks and Pathway Analysis
Part II Learning, Classification, and Clustering
Part III Data Mining and Knowledge Discovery
Part IV Protein: Structure, Function, and Interaction
Part V Motifs, Sites, and Sequences Analysis

Part I of the proceedings contains six chapters on "Bio-Molecular Networks and Pathway Analysis." Rahman et al. propose a fast agglomerative clustering method for protein complex discovery. A new criterion is introduced that combines an edge clustering coefficient and an edge clustering value, allowing us to decide when a node can be added to the current cluster. Maduranga et al. use the well-known random forest method to predict GRNs. The problem of inferring GRNs from (limited) time-series data is recast as a number of regression problems, and the random forest approach is used here to fit a model to this. Winterbach et al. evaluate how well topological signatures in protein interaction networks predict protein function. They compare several complex signatures and their own simple signature. They find that network topology is only a weak predictor of function and the simple signature performs on par with the more sophisticated ones. De Ridder et al. propose an approach for identifying putative cancer pathways. This approach relies on expression profiling tumors that are induced by retroviral insertional mutagenesis. This provides the opportunity to search for associations between tumor-initiating events (the viral insertion sites) and the consequent transcription changes, thus revealing putative regulatory interactions. An important advantage is that the selective pressure exerted by the tumor growth is exploited to yield a relatively small number of loci that are likely to be causal for tumor formation. Ochs et al. apply outlier statistics, gene set analysis, and top scoring pair methods to identify deregulated pathways in cancer. Analysis of the results on pediatric acute myeloid leukemia data indicate the effectiveness of the proposed methodology. Pizzuti et al. present some variants of RNSC (restricted neighborhood search clustering) for prediction of protein complexes that are based on new score functions and evolutionary computation. It is shown via computational experiments that the proposed methods have better prediction accuracies (in F-measure) than the basic RNSC algorithm.

Part II of the proceedings contains three chapters on "Learning, Classification, and Clustering." Marchiori addresses a limitation of the RELIEF feature weighting algorithm that maximizes the sample margin over the entire training set, or the sum of the possibly competing feature weights. Her work proposes, instead, a conditional weighting algorithm (CCFW) and classifier (CCWNN) to improve feature weighting and classification. Mundra et al. propose a sample selection criterion using a modified logistic regression loss function and a backward elimination based gene ranking algorithm. On the basis of the classifier margin for sample points, points on or within the margin are more important than those outside, the sample selection criterion based on T-score is proposed. Li et al. describe a generalization of sparse matrix factorization (SMF) algorithms and showcase a few very concisely described applications in bioinformatics. The main merit of the work is the fact that a unified representation for SMF algorithms is proposed, as well as an optimization algorithm to solve this problem.

Part III of the proceedings contains six chapters on "Data Mining and Knowledge Discovery." Hsu et al. consider prediction of RNA secondary structure in the "triple helix" setting for which they argue existing methods are inadequate. Their approach uses a Simple Tree Adjoining Grammar (STAG) coupled

with maximum likelihood estimation (MLE), implemented via an efficient dynamic programming formulation. Higgs et al. present an algorithm for generating near-native protein models. It combines a fragment feature-based resampling algorithm with a local optimization method that performed best, for protein structure prediction (PSP), among a set of five optimization techniques. Computational experiments show that the use of local optimization is beneficial in terms of both RMSD and TM score. Spirov et al. discuss a method for transformation of variables, in order to normalize *Drosophila oocyte* images acquired via confocal microscopy. The paper describes an interesting problem, namely, the experimental determination of intrinsic *Drosophila* embryo coordinates, and proposes an approach using evolutionary computation by genetic algorithms. Rezaeian et al. propose a novel and flexible hierarchical framework to select discriminative genes and predict breast tumor subtypes simultaneously. Dai et al. tackle an important problem in drug-target interaction research and present an interesting application of machine learning methods to the analysis of drugs. Gritsenko et al. make an adaptation of their previously developed protocol for building and evaluating predictors, in order to introduce a framework that enables forward engineering in biology. An experimental test is performed in the biological field of codon optimization and the results obtained are comparable with those produced by the reference tool JCat.

Part IV of the proceedings contains six chapters on "Protein: Structure, Function, and Interaction." Xiong et al. propose an active learning-based approach for protein function prediction. The novelty of the proposal is the use of a preprocessing phase that uses spectral clustering before selecting candidates for labeling with graph centrality metrics. Experimental results show that clustering reveals a valid pre-processing step for the active learning method. Gehrmann et al. address the problem of integrating multiple sources of evidence to predict protein functions. The paper proposes to use a conditional random field (CRF) to represent protein functions as random variables to be predicted and different sources of evidence as conditioning variables. Inference and learning algorithms based on MCMC are described and the proposed method is applied to a yeast dataset. Dehzangi et al. describe a new approach to protein fold recognition, a problem that has been widely studied over the past decade. The main contribution is the proposal of a new set of global protein features based on evolutionary consensus sequences and predicted secondary structure, and local features based on distributions and auto covariances of these features over segments. An RBF SVM using these features is applied to two benchmark datasets in an extensive comparison with a number of existing methods and is demonstrated to work well. Dehzangi et al. present a novel approach to using features extracted from the position specific scoring matrix (PSSM) to predict the structural class of a protein. The authors propose two new sets of features: a global one based on the consensus sequence of a PSSM and a local one that takes the auto-covariance in sequence segments into account. The features extracted are used to train an RBF SVM and are shown to lead to good results (better than other state-of-the-art algorithms) on two benchmarks. Chiu et al. discuss a new method for detecting

associated sites in aligned sequence ensembles. The main idea is derived from the concept of granular computing, where information is extracted at different levels of granularity or resolution. The experimentation was focused on p53 and it has been demonstrated that the extracted association patterns are useful in discovering sites with some structural and functional properties of a protein molecule. Tung presents a new method for predicting the potential hepatocarcinogenicity of non-genotoxic chemicals. The proposed method based on chemical–protein interactions and interpretable decision tree is compared with other data-mining approaches and shows very good performances in both accuracy and simplicity of the found model.

Part V of the proceedings contains four chapters on "Motifs, Sites, and Sequences Analysis." Pathak et al. present an algorithm that exploits structural information for reducing false positives in motifs prediction. They tested the validity of the algorithm using the minimotifs stored in the MnM database. Lacroix et al. present a workflow for the prediction of the effects of residue substitution on protein stability. The workflow integrates eight algorithms that use delta-delta-G as a measure of stability. The workflow is designed to populate the online resource SPROUTS. A use case of the workflow is presented using the PDB entry 1enh. Malhotra et al. present an algorithm for inferring haplotypes of virus populations from k-mer counts obtained from next-generation sequencing (NGS) data. The algorithm takes as input read counts for a set of k-mers and produces as output a predicted number of haplotypes, their relative frequencies and, for reads covering SNPs, can assign reads to a haplotype. The novel feature of the algortihm is that it does not rely on having a reference genome. The authors report that it performs well on synthetic data compared with the existing algorithm ShoRAH, which relies on a reference genome. Comin et al. discuss and improve the Entropic Profile method introduced in the literature for detecting conservation in genome sequences. The authors propose a linear-time linear-space algorithm that captures the importance of given regions with respect to the whole genome, suitable for large genomes and for the discovery of motifs with unbounded length.

Many have contributed directly or indirectly toward the organization and success of the PRIB 2013 conference. We would like to thank all the individuals and institutions, especially the authors for submitting the papers and the sponsors for generously providing financial support for the conference. We are very grateful to IAPR for the sponsorship. Our gratitude goes to the Nice Sophia Antipolis University, Nice, France, and IAPR (International Association for Pattern Recognition) Bioinformatics Technical Committee (TC-20) for supporting the conference in many ways.

We would like to express our gratitude to all PRIB 2013 International Program Committee members for their objective and thorough reviews of the submitted papers. We fully appreciate the PRIB 2013 Organizing Committee for their time, efforts, and excellent work. We would also like to thank the Nice Sophia Antipolis University for hosting the symposium and providing technical support. We sincerely thank the EDSTIC doctoral school for providing grants to

a number of students attending the conference. We also thank "Region PACA" and the University of Salerno (Italy) for partially funding the invited speakers.

Last, but not least, we wish to convey our sincere thanks to Springer for providing excellent professional support in preparing this volume.

June 2013

Alioune Ngom
Enrico Formenti
Jin-Kao Hao
Xing-Ming Zhao
Twan van Laarhoven

PRIB 2013 Organization

PRIB General Chair

Alioune Ngom	University of Windsor, Canada

CIBB-PRIB General Chair

Enrico Formenti	Nice Sophia Antipolis University, France

Program Co-chairs

Jin-Kao Hao	University of Angers, France
Xing-Ming Zhao	Tongji University, China

Publicity Chair

Gilles Bernot	Nice Sophia Antipolis University, France

Local Arrangements Chair

Jean-Paul Comet	Nice Sophia Antipolis University, France

Publication Chair

Twan van Laarhoven	Radboud University, The Netherlands

PRIB Steering Committee

Raj Acharya	PennState, USA
Shandar Ahmad	Tokyo Tech NIBIO, Japan
Madhu Chetty	Monash University, Australia
Tom Heskes	Radboud University of Nijmegen, The Netherlands
Visakan Kadirkamanathan	University of Sheffield, UK
Elena Marchiori	Radboud University of Nijmegen, The Netherlands
Jagath C. Rajapakse	Nanyang Technological University, Singapore
Marcel Reinders	Delft University of Technology, The Netherlands
Dick de Ridder	Delft University of Technology, The Netherlands
Guido Sanguinetti	University of Edinburgh, UK
Jun Sese	Tokyo Tech NIBIO, Japan

Program Committee

Raj Acharya	Pennsylvania State University, USA
Shandar Ahmad	National Institute of Biomedical Innovation, Osaka
Tatsuya Akutsu	Kyoto University, Japan
Sahar Al Seesi	University of Connecticut, USA
Hisham Al-Mubaid	University of Houston, USA
Ashish Anand	Institut Pasteur Paris, France
Kiyoko Aoki-Kinoshita	Soka University, Japan
Wendy Ashlock	University of Guelph, Canada
Francisco Azuaje	Public Research Centre for Health, UK
Jaume Bacardit	University of Nottingham, UK
Pedro Ballester	Cambridge University, UK
Sanghamitra Bandyopadhyay	Indian Statistical Institute, India
Gilles Bernot	Nice Sophia Antipolis University, France
Chengpeng Bi	University of Missouri, UK
Sebastian Böcker	Friedrich Schiller University of Jena, Germany
Conrad Burden	Australian National University, Australia
David Cairns	University of Stirling, UK
Rachel Cavill	Imperial College London, UK
Frederic Cazals	INRIA Sophia, France
Keith C.C. Chan	The Hong Kong Polytechnic University, China
Kuo-Sheng Cheng	National Cheng Kung University, Taiwan
Francis Chin	The University of Hong Kong, China
Sung-Bae Cho	Yonsei University, Korea
Young-Rae Cho	Baylor University, USA
Dominique Chu	University of Kent, UK
Pau-Choo Chung	National Cheng Kung University, Taiwan
Steven Corns	Missouri University of Science and Technology, USA
Sanjoy Das	Kansas State University, USA
Dick De Ridder	Delft University of Technology, The Netherlands
Jeroen De Ridder	Delft University of Technology, The Netherlands
Tjeerd Dijkstra	Radboud University, The Netherlands
Federico Divina	Pablo de Olavide University, Spain
Beatrice Duval	University of Angers, France
Mansour Ebrahimi	University of Qom, Iran
Esmaeil Ebrahimie	Shiraz University, Iran
Richard Edwards	University of Southampton, UK
Antonino Fiannaca	ICAR-CNR, Italy
Maurizio Filippone	University of Glasgow, UK

Christoph M. Friedrich	University of Applied Science and Arts, Germany
Rosalba Giugno	University of Catania, Italy
Robin Gras	University of Windsor, Canada
Michael Gromiha	IIT Madras, India
Michael Hahsler	Southern Methodist University, USA
Jennifer Hallinan	Newcastle University, UK
Xiaoxu Han	University of Iowa, USA
Timothy Havens	University of Missouri, USA
Morihiro Hayashida	Kyoto University, Japan
David Hecht	Southwestern College, USA
Md Tamjidul Hoque	Griffith University, Australia
Sheridan Houghten	Brock University, Canada
Liang-Tsung Huang	Mingdao University, Taiwan
Seiya Imoto	University of Tokyo, Japan
Zhenyu Jia	University of California Irvine, USA
Colin Johnson	University of Kent, UK
Laetitia Jourdan	INRIA, France
David Juedes	Ohio University, USA
Giuseppe Jurman	Fondazione Bruno Kessler, Italy
R. Krishna Murthy Karuturi	Genome Institute of Singapore, Singapore
Marta Kasprzak	Poznan University of Technology, Poland
Yuki Kato	Nara Institute of Science and Technology, Japan
Tsuyoshi Kato	University of Tokyo, Japan
Nawaz Khan	Middesex University, UK
Seyoung Kim	Carnegie Mellon University, USA
Kyung Dae Ko	Howard University, USA
Ziad Kobti	University of Windsor, Canada
Tetsuji Kuboyama	Gakushuin University, Japan
Lukasz Kurgan	University of Alberta, Canada
Zoe Lacroix	Arizona State University, USA
Yifeng Li	University of Windsor, Canada
Xiaoli Li	Institute for Infocomm Research, Singapore
Wingning Li	University of Arkansas, USA
Feng Lin	Nanyang Technological University, Singapore
Frédérique Lisacek	Swiss Institute of Bioinformatics, Switzerland
Chunmei Liu	Howard University, USA
Xuejun Liu	Nanjing University of Aeronautics and Astronautics, China
Weiguo Liu	Nanyang Technological University, Singapore
Huaien Luo	Genome Institute of Singapore, Singapore
Hiroshi Matsuno	Yamaguchi University, Japan
Ken Mcgarry	University of Sunderland, UK

Vasilis Megalooikonomou	Temple University, USA
Mariofanna Milanova	University of Arkansas at Little Rock, USA
Aleksandar Milosavljevic	Baylor College of Medicine, USA
Perry Moerland	Academic Medical Center, The Netherlands
Jason Moore	Dartmouth College, USA
Vadim Mottl	Computing Center of the Russian Academy of Sciences, Russia
Piyushkumar Mundra	Nanyang Technological University, Singapore
Julio Cesar Nievola	Pontifícia Universidade Católica do Paraná, Brazil
Michael Ochs	Johns Hopkins University, USA
Carlotta Orsenigo	Politecnico di Milano, Italy
Deanna Petrochilos	University of Washington, USA
Thang Pham	VU University Medical Center, The Netherlands
Clara Pizzuti	CNR-ICAR, Italy
Kiran Sree Pokkuluri	Venkata Chalamayya Engineering College, India
Gianfranco Mihele Maria Politano	Politecnico di Torino, Italy
Beatriz Pontes	University of Seville, Spain
Sanguthevar Rajasekaran	University of Connecticut, USA
Jean-Michel Richer	University of Angers, France
Ricardo Rizzo	National Research Council of Italy, Italy
Miguel Rocha	University of Minho, Portugal
Katya Rodriguez-Vazquez	IIMAS-UNAM, Mexico
Eduardo Rodriguez-Tello	Cinvestav-Tamaulipas, Mexico
Luis Rueda	University of Windsor, Canada
Gonzalo Ruz	Universidad Adolfo Ibáñez, Chile
Yvan Saeys	Ghent University, Belgium
Hiroto Saigo	Kyushu Institute of Technology, Japan
Taro L. Saito	University of Tokyo, Japan
Roberto Santana	Technical University of Madrid, Spain
Bertil Schmidt	University of Mainz, Germany
Christian Schönbach	Kyushu Institute of Technology, Japan
Huseyin Seker	De Montfort University, UK
Masakazu Sekijima	Tokyo Institute of Technology, Japan
Jun Sese	Tokyo Institute of Technology, Japan
Anne Siegel	IRISA – CNRS, France
Luciano Silva	Universidade Federal do Parana, Brazil
Evangelos Simeonidis	Luxembourg Centre for Systems Biomedicine, Luxembourg
Alexander Spirov	SUNY at Stony Brook, USA
Bela Stantic	Griffith University, Australia
Gregor Stiglic	University of Maribor, Slovenia

Chrysostomos Stylios	Technological Education Institute of Epirus, Greece
Wing-Kin Sung	Nuational University of Singapore, Singapore
Kenji Suzuki	The University of Chicago, USA
Marta Szachniuk	Polish Academy of Sciences, Poland
Sandor Szilagyi	Sapientia-Hungarian Science University of Transylvania, Hungary
Roberto Tagliaferri	University of Salerno, Italy
Shyh Wei Teng	Monash University, Australia
Spencer Thomas	University of Surrey, UK
Renato Tinós	University of Sao Paulo, Brazil
Anna Tramontano	University of Rome La Sapienza, Italy
Herbert Treutlein	Computist Bio-Nanotech, Australia
Herbert H. Tsang	Simon Fraser University, Canada
Alexey Tsymbal	Siemens AG, Germany
Chun-Wei Tung	National Chiao Tung University, Taiwan
Marcel Turcotte	University of Ottawa, Canada
Alfonso Urso	Consiglio Nazionale delle Ricerche, Italy
Jean-Philippe Vert	Ecole des Mines de Paris, France
Jiri Vohradsky	Institute of Microbiology ASCR, Czech Republic
Junbai Wang	Radium Hospital, Norway
Lusheng Wang	City University of Hong Kong, Hong Kong
Haixuan Yang	Royal Holloway University of London, UK
Junichiro Yoshimoto	Okinawa Institute of Science and Technology, Japan
Yanqing Zhang	Georgia State University, USA
Daming Zhu	Shandong University, China

Table of Contents

Part III: Data Mining and Knowledge Discovery

Part IV: Protein: Structure, Function, and Interaction

A Fast Agglomerative Community Detection Method for Protein Complex Discovery in Protein Interaction Networks

Mohammad S. Rahman and Alioune Ngom

School of Computer Sciences, 5115 Lambton Tower, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{rahman1v,angom}@uwindsor.ca

Abstract. Proteins are known to interact with each other by forming protein complexes and in order to perform specific biological functions. Many community detection methods have been devised for the discovery of protein complexes in protein interaction networks. One common problem in current agglomerative community detection approaches is that vertices with just one neighbor are often classified as separate clusters, which does not make sense for complex identification. Also, a major limitation of agglomerative techniques is that their computational efficiency do not scale well to large protein interaction networks (PINs). In this paper, we propose a new agglomerative algorithm, FAC-PIN, based on a local premetric of relative vertex-to-vertex clustering value and which addresses the above two issues. Our proposed FAC-PIN method is applied to eight PINs from different species, and the identified complexes are validated using experimentally verified complexes. The preliminary computational results show that FAC-PIN can discover protein complexes from PINs more accurately and faster than the HC-PIN and CNM algorithms, the current state-of-the-art agglomerative approaches to complex prediction.

1 Introduction

Proteins are known to interact with each other by forming complexes. Each such complex performs an independent and discrete biological function through the interactions of its member proteins [9]. Single proteins may also participate in more than one complex. Protein complexes correspond to *modules*, which are dense subgraphs within PINs, and hence, they can be discovered by appropriate graph clustering approaches. Generally speaking, modules in PINs refer to highly connected subgraphs which have more internal edges than external edges. Many definitions of modules have been proposed in literature [16], and consequently different community detection algorithms have been proposed based on these different definitions.

Module detection in PINs is a computationally hard task and conventional clustering algorithms are not well suited for this task [15, 20]. Efficient, accurate, robust, and scalable methods are therefore required for mining large PINs. There are generally three classes of modules detection approaches: 1) those based on finding *cliques*, which are fully connected subnetworks [11, 17]; 2) those based on detecting dense subnetworks [1, 2], not necessarily cliques; and 3) those based on uncovering the hierarchical

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 1–12, 2013.

organization of modules within PINs [8, 12]. Clique techniques are not quite scalable to large PINs and the identified modules are too strict in the biological sense of modules since proteins participating in a complex may not all interact with each other. Current density-based algorithms commonly misclassify proteins with low degree into small clusters which could be merged to core protein clusters [13]. Moreover, many biologically meaningful modules are ignored due to their low topological connectivity [13]. Hierarchical clustering methods based on global metric over nodes or edges, such as betweenness centralities, are very time-consuming, and thus do not scale well to large PINs. The few hierarchical approaches based on local metric also have the common problem of classifying vertices with degree one in separate clusters, which does not make sense biologically.

In this paper, we propose a fast agglomerative clustering technique, FAC-PIN, which addresses the limitations discussed above for hierarchical algorithms. FAC-PIN is based on a local premetric of relative vertex clustering value for clustering PINs in a hierarchical manner.

The rest of the paper is organized as follow. In Section 2, we discuss a few hierarchical algorithms to which FAC-PIN is based. Section 3 introduces our proposed method. Computational experiments and discussions of results are given in Section 5 before we conclude with possible directions of research.

2 Related Works

Many hierarchical clustering approaches (both agglomerative and divisive techniques) have been introduced in literature, since the original publication of Girvan and Newman in [7] for clustering networks. See the excellent survey on graph clustering algorithms in [5]. Thus, we will present only the few methods that are directly related to our proposed agglomerative approach.

An effective agglomerative technique for clustering large networks was first proposed by Girvan and Newman in [7]. The Girvan and Newman (GN) algorithm first computes the edge-betweenness centrality value of each edge; this is a global metric over the edges and is defined as the number of shortest paths containing a given edge. Then, GN subsequently sort and then remove edges with large betweenness values in an iterative manner and in order to detect the communities; since such edges correspond to *bridges* connecting two modules whereas low-betweenness edges are internal to modules. To increase the computational speed of GN, Clauset et al. [4] made a simple but non-trivial modification in the computation of the value of the modularity function used in GN. Luo et al. [13] defined the concept of the degree of a subnetwork S as the number the of edges containing one endpoint inside S and the other endpoint outside S. The degree of subnetworks was used along with the edge-betweenness values to devise an agglomerative method for module discovery. Li et al. [12] developed a fast agglomerative approach for community detection based on a global centrality measure, the *vertex clustering coefficient*; which is defined as the ratio of the number of edges between the neighbors of a given vertex v and the total number of possible edges in that neighborhood, it measures the degree of completeness of the subnetwork defined by v and its neighbors [6]. Radicchi et al. [16] designed an agglomerative technique based

on the clustering coefficient of an edge; the *edge clustering coefficient* extends the vertex clustering coefficient and is a global measure defined as the number of triangles to which a given edge $e = (u, v)$ belongs to, divided by the number of triangles that might potentially include (u, v). That is:

$$C^{(3)}_{u,v} = \frac{Z^{(3)}_{u,v}}{\min\{(k_u - 1), (k_v - 1)\}}, \tag{1}$$

where, k_a is the degree of a vertex a, $Z^{(3)}_{u,v}$ is the number of triangles containing edge (u, v), and $\min\{(k_u - 1), (k_v - 1)\}$ is the maximal possible number of triangles containing (u, v). This coefficient has been further generalized to higher-order cycles, $C^{(k)}_{u,v}$, such as squares for $k = 4$, $C^{(4)}_{u,v}$. Edges contained in few or no triangles have low clustering coefficients, and hence, correspond to *bridges* connecting two clusters. The edge clustering coefficient assumes the existence of cycles of length k in a network; which is problematic since a network can have many cycles of different lengths and the length distribution is unknown (e.g., there may be very few or very many short-length cycles). For this reason, Wang et al. [19] defined a local metric over the edges, the *edge clustering value*, which is not based on cycles but on the common neighbors of the two endpoints of edge (u, v). The edge clustering value is defined as:

$$ECV(u, v) = \frac{|N_u \cap N_v|^2}{|N_u| \times |N_v|}, \tag{2}$$

where, N_a is the set of neighbors of a vertex a and its cardinality is defined as $|N_a|$. Here, endpoints vertices of an edge (u, v) with a larger clustering value are more likely to be in the same cluster. Using the edge clustering value, Wang et al. [19] devised an agglomerative technique, the HC-PIN algorithm, for discovering modules of a PIN and which is faster and more accurate than current hierarchical algorithms for network clustering.

In the following section, we introduce a new measure, the *relative vertex-to-vertex clustering value*, which is a premetric combining the ideas behind the vertex clustering coefficient, the edge clustering coefficient, and the edge clustering value. Our analysis of this measure will be based on the *weak sense* definition of a community (i.e., a module); that is: a subgraph S is a community in a weak sense if the sum of all degrees within S (i.e., sum of its internal edges) is larger than the sum of all degrees toward the rest of the network (i.e., sum of its external edges) [16].

3 Relative Vertex-to-Vertex Clustering Value

The edge clustering value, $ECV(u, v)$, used in HC-PIN [19], is a similarity metric between the two vertices u and v of an edge (u, v) and which, roughly speaking, tells how likely u and v lie in the same module (i.e., cluster). This is also true with the edge clustering coefficient, $C^{(3)}_{u,v}$, of [16]. However, in complex networks following the power law (i.e., scale-free networks), it is reasonable to assume that the likelihood of a vertex u to lie in the same module as v (or, to lie in the module containing v), is not

equal to the likelihood of v to lie in the module containing u. This assumption stems from the principle of *preferential attachment* in scale-free networks which states that a new node u is likely to *attach* to a high-degree node v than to a low degree node. This is not reciprocal, and hence, clearly suggesting that the likelihood is not symmetric and that it is larger for u to be in a cluster with v than for v to be in cluster with u (if we assume that v is a high-degree node). The similarity metrics $ECV(u, v)$ and $C^{(3)}_{u,v}$ treat equally both endpoints of edges (u, v) irrespective of their degrees. Also, another issue is that both $ECV(u, v)$ and $C^{(3)}_{u,v}$ require vertices u and v be connected by an edge. This requirement is quite restrictive and we aim to extend to the case in which pair (u, v) is not an edge while still being able to decide if both vertices are in the same cluster. Finally, as stated earlier in previous section, current hierarchical approaches have the common problem of classifying low-degree vertices (peripheral to dense subnetwork modules) into separate clusters rather than merging them with their neighboring modules. In the following paragraph, we present a new measure which aims to address these issues.

Let N_a be the set of neighbors of vertex a in an undirected graph $G = (V, E)$. We define $N^+_a = N_a \cup \{a\}$ as the neighbor set of a augmented with a itself. Given two vertices u and v, we define the clustering value of u relative to v as:

$$R(u \dashrightarrow v) = \frac{|N^+_u \cap N^+_v|}{|N^+_u|} \tag{3}$$

$R(u \dashrightarrow v)$ is a premetric that ranges from 0 to 1; that is, it is a measure which does not satisfy the axiom of symmetry and the triangle inequality but satisfies the axioms of self-similarity and minimality. A vertex u with a larger clustering value given another vertex v is more likely to lie in the cluster containing v. In the following $C(a)$ denotes the cluster containing a given vertex a, and we assume that $C(a)$ satisfies the *weak sense* definition of a community [16] (we use the term ws-cluster, hereafter). The following describe the properties of $R(u \dashrightarrow v)$.

Given an edge (u, v), $R(u \dashrightarrow v)$ is maximal (i.e. equals 1) if and only if $|N^+_u| = |N^+_u \cap N^+_v|$. There are two cases achieving the maximum given edge (u, v): (i) when u has degree one; and (ii) when both u and v have the same degree and $|N^+_u| = |N^+_v|$ that is, they have the same neighbors. In either case, If sub-network $C(v)$ (respectively, the induced sub-network of G for subset N^+_v) is a ws-cluster then $\{u\} \cup C(v)$ (respectively, $\{u\} \cup N^+_v$) is a also a ws-cluster.

Given an edge (u, v), $R(u \dashrightarrow v)$ is minimal when u is the highest degree vertex in G and v has degree 1; that is, $R(u \dashrightarrow v) = \frac{2}{1+deg(u,G)}$ and $deg(u, G)$ is maximal. In such case, $R(v \dashrightarrow u)$ is maximal (i.e. equals 1), and hence, $C(u) \cup \{v\}$ (respectively, $N^+_u \cup \{v\}$) is a ws-cluster if $C(u)$ (respectively, N^+_u) is a ws-cluster.

Given an edge (u, v), assume the degrees of vertices u and v in G are such that $deg(u, G) = deg(v, G) = d$ is maximal and that u and v do not share any other neighbors. Then, we have $R(u \dashrightarrow v) = R(v \dashrightarrow u) = \frac{2}{1+d} \leq 0.5$ assuming $d \geq 3$. In this case, $\{u\} \cup C(v)$ (or N^+_v) is not a ws-cluster, and, $\{v\} \cup C(u)$ (or N^+_u) is not a ws-cluster. Consider the induced subgraph of G on $N^+_u \cup N^+_v$, we define the *local betweenness value* of edge (u, v) as the percentage of paths from vertices in $N_u \setminus N_v$ to vertices in $N_v \setminus N_u$ going through edge (u, v). Given the number of

common neighbors between u and v, $|N_u \cap N_v|$, the local betweenness of edge (u, v) is thus $l(u, v) = 100 \cdot \frac{1}{|N_u \cap N_v|+1}$. Given two connected high-degree vertices u and v, the local edge betweenness value $l(u, v)$ increases when $|N_u \cap N_v|$ decreases, and hence, it corresponds to when both $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ values are small at the same time. Edges with high local betweenness values are edges connecting two clusters, and therefore, vertices u and v should not lie in the same cluster.

Finally, our relative vertex clustering values implements the ideas behind the edge clustering coefficient, $C^{(k)}_{u,v}$, of [16], since for a given vertex v and a neighbor u the number of triangles given edge (u, v) is exactly $|N_u \cap N_v|$; and u will be included into $C(v)$ whenever most of the neighbors of u (excluding v) are in $N_u \cap N_v$. This is also true even when (u, v) is not an edge; in such case, $|N_u \cap N_v|$ relates to the number of squares containing vertices u and v. On the other hand, we break through the limitations of [16] as in the edge clustering value, $ECV(u, v)$ of [19], by not assuming the existence of closed loops in a networks, such as triangles or high-order loops. The relative vertex clustering value $R(u \dashrightarrow v)$ also improves $ECV(u, v)$ since neighbors u of v which have most of their neighbors forming a triangle with v are selected for inclusion in $C(v)$. Searching for vertices u which form a cluster with v is also more efficient than searching for edges (u, v) that makes a cluster since the number of edges is larger than the number of vertices in dense subgraphs.

In summary, the values $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ for edge (u, v) can be used as a quick test for deciding whether u (respectively, v) should be merged with the cluster $C(v)$ (respectively, $C(u)$) such that $\{u\} \cup C(v)$ (respectively, $\{v\} \cup C(u)$) remains a ws-cluster.

4 The FAC-PIN Algorithm

Our proposed fast agglomerative clustering algorithm for protein interaction networks, FAC-PIN in Algorithm 1, goes as follows. Given a PIN $G = (V, E)$, we initially consider each vertex as a singleton cluster, and sort the vertices $v \in V$ in decreasing order of their degrees $deg(v, G)$ in G. Then, in an iterative manner, we select the next highest-degree vertex v from the sorted list, and compute the values $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ for each neighbor u of v, and then decide depending on these two values and a threshold $\alpha, 0 \leq \alpha \leq 1$, whether u should be included in $C(v)$ or not.

In the FAC-PIN algorithm, a neighbor u of vertex v is added to the current $C(v)$ when the majority of the neighbors of u are in $N_u \cap N_v$, that is when: 1) $R(u \dashrightarrow v) = 1$, in which case either u has degree 1, or u and v have the same degree and the same set of neighbors; 2) $R(u \dashrightarrow v) > R(v \dashrightarrow u) > \alpha$, in which case u have smaller degree than v and most of the neighbors of u are in the intersection; and 3) $R(u \dashrightarrow v) = R(v \dashrightarrow u)$ and the size of the intersection is larger than the total set of neighbors of u and v which are not in the intersection.

Computational Complexity of FAC-PIN: Let $n = |V|$ be the number vertices, $m = |E|$ be the number of edges, and $\bar{d}$ be the average degree of all vertices, that is $\bar{d} = \frac{1}{n}\sum_{v \in V} deg(v, G)$. The complexity of sorting the vertices by their degree is $O(n)$ by using the *counting sort* method, and the complexity of computing the partition after

Algorithm 1. The *FAC-PIN* Algorithm

Input: $G = (V, E)$: undirected PIN graph
α: threshold parameter
Output: $P_k = \{C_1, \ldots, C_k\}$: identified collection of modules
{**Initialization phase**}
for every $v_i \in V$ **do**
$C(v_i) \leftarrow \{ \{v_i\}, \emptyset \}$; {*each vertex is a singleton cluster*}
end for
Sort all vertices to a priority-queue H in non-increasing order of their degrees;
{**Community detection phase**}
repeat
$v \leftarrow H$; {*select next highest-degree vertex in* H}
for all $u \in N_v$ not yet merged into a cluster **do**
if $[R(u \dashrightarrow v) = 1]$ Or $[R(u \dashrightarrow v) > R(v \dashrightarrow u) > \alpha]$ **then**
$C(v) \leftarrow C(v) \cup \{ \{u\}, \{(u, v)\} \}$;
$C(u) \leftarrow C(v)$;
else
if $[R(u \dashrightarrow v) = R(v \dashrightarrow u)]$ And $[deg(u, G) + deg(v, G) - 1 \leq |N_u \cap N_v|]$ **then**
$C(v) \leftarrow C(v) \cup \{ \{u\}, \{(u, v)\} \}$;
$C(u) \leftarrow C(v)$;
end if
end if
end for
until $H = \emptyset$
$U \leftarrow V$;
$i \leftarrow 1$;
{**Compute the partition** P_k}
while $U \neq \emptyset$ **do**
$v \leftarrow$ randomly select a vertex from U;
$C_i \leftarrow C(v)$;
$U \leftarrow U \setminus \{u \mid C(u) = C(v)\}$;
$i \leftarrow i + 1$;
end while
return $P_k \leftarrow \{C_1, \ldots, C_k\}$;
Evaluate modularity $Q(P_k)$ of partition $P_k = \{C_1, \ldots, C_k\}$;

the community detection phase is also $O(n)$. Let the maximum node degree in G be $d_{\max} = \max_{v \in V} deg(v, G)$. The complexity of computing $R(u \dashrightarrow v)$ given vertices u and v in the "for-loop" of FAC-PIN is $O(d_{\max})$. The complexity of the "for-loop" is then $O(d_{\max}^2)$, and hence, the total complexity of the "repeat-loop" (and thus of FAC-PIN) is $O(nd_{\max}^2) \ll O(n^3)$. Since PINs are power-law networks then the majority of the proteins interact with only very few proteins, and thus the average degree $\bar{d}$ is generally small and can be considered a constant [19]; that is, we can use $\bar{d}$ as the principal variable for measuring the complexity of community detection methods. As such, then the complexity of FAC-PIN is $O(n\bar{d}^2) \ll O(nd_{\max}^2) \ll O(n^3)$. The complexity of the HC-PIN algorithm of [19] is $O(m\bar{d}^2)$ and is larger than that of FAC-PIN since

$n \lll m$ in PINs. We note that HC-PIN is currently the fastest hierarchical method described in literature for clustering PINs, as far as we know.

5 Computational Experiments and Discussions

We have carried out several computational experiments on the PIN data of eight different species using our proposed FAC-PIN algorithm. For each PIN, we performed the following steps sequentially: (1) we arbitrarily set the threshold parameter, α in FAC-PIN, to values $0.5, 0.25, 0.125, 0.0625$ and 0.03125, (2) applied FAC-PIN to the given PIN, with each of these values, (3) evaluated the modularity (i.e., the goodness) of the resulting partition P_k for a value α, and finally (4) we reported the partition result for the value α (among all given values) which gives the best modularity value. The PINs and the modularity evaluation functions are discussed below.

PIN Data: The PINs data of eight different species were obtained from the PINALOG site[1] and the BioGRID database[2]. The eight species given along with their number of proteins and interactions in parenthesis are: *E. coli* (2817, 13841), *D. melanogaster* (Fruit fly, 8366, 25611), *A. thaliana* (Flowering plant, 2651, 5236), *M. musculus* (House mouse, 2888, 4372), *H. sapiens* (Human, 8994, 34935), *R. norvegicus* (Street rat, 1148, 1307), *C. elegans* (Round worm, 4303, 7747), and *S. cerevisiae* (Bakers yeast, 5672, 49830). In all these PINs, the number of edges is much larger than the number of vertices.

Modularity Functions: Given a clustering result (i.e. a partition) $P_k = \{C_1, \ldots, C_k\}$ with k clusters, we used the popular modularity function introduced by Newman and Girvan [4], defined as

$$Q(P_k) = \sum_{i=1}^{k} (e_{ii} - a_i^2), \tag{4}$$

where, e_{ii} is the fraction of edges with both end vertices in the same community i, and a_i is the fraction of edges with at least one end vertex in community i. Larger values of Q correspond to more distinct community structures in PINs. Though Q is widely used, it is known to have serious limitations which has been discussed at length in [5]. The second partition scoring function we used has been introduced in [10] and is defined as

$$w\text{-}\log\text{-}v(P_k) = \sum_{i=1}^{k} (e_{ii} - \log a_i). \tag{5}$$

Function $w\text{-}\log\text{-}v$ allows for more diverse cluster sizes than function Q, and smaller values corresponds to better modularity structures.

[1] `http://www.sbg.bio.ic.ac.uk/~pinalog/downloads.html`
[2] `thebiogrid.org`

Computational Results: As said above, we applied FAC-PIN many times on a given PIN data but with a different threshold value α in each run, then evaluated the resulting partition for that value α, and then retained the best partition P_k obtained for the PIN among all values α. The best partition is that which has the best modularity value. In order to study and compare the performance of FAC-PIN, we downloaded the CNM code from `http://cs.unm.edu/~aaron/research/fastmodularity.htm`[4] and implemented the HC-PIN algorithm [19]. The HC-PIN and CNM methods were applied on the same PIN data as the FAC-PIN approach. For HC-PIN, we set the two parameters λ and s as in [19] (CNM has no parameters). The modularity results of the three methods are given in Tables 1 and 2, and their running times are shown in Table 3. The PINs are sorted in increasing order of their number of proteins (that is, *Street rat*'s PIN being the smallest is on first column and *Human*'s PIN being the largest is on the last column).

Table 1. Q results of *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	0.7897	0.9422	0.1492	0.7644	0.7484	0.5110	0.6486	0.7827
CNM	0.5457	0.7861	0.0587	0.4781	0.4057	0.1412	0.3116	0.2858
HC-PIN	0.4502	0.7819	0.0023	0.5015	0.2928	0.0387	0.0086	0.0126

Table 2. w-$\log$-v results of *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	-2.252	-3.603	-0.262	-2.634	-2.094	-0.521	-1.517	-1.941
CNM	-1.699	-2.866	-0.192	-1.530	-1.819	-0.481	-1.233	-1.269
HC-PIN	-1.558	-3.071	-0.019	-1.805	-1.809	-0.028	-0.072	-0.113

As we see in both Tables 1 and 2, FAC-PIN outperformed both the HC-PIN and CNM methods in all given PINs. We note that as the size of the PINs increases, in terms of either the number of proteins or the number of interactions, the difference between the performances of FAC-PIN and HC-PIN (or CNM) also increase greatly. This is also true in Table 3 showing the execution times, in seconds, of the three algorithms. Clearly FAC-PIN is much faster than the other two methods, and again, the difference in performance increases as either the number of proteins or the number of interactions increases. All experiments were performed on an Intel machine (Core TM i7-2600, 3.400 GHz, CPU with 8 GB RAM).

Table 3. Time results *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	1.00	4.77	3.66	7.44	22.25	25.12	54.85	72.59
CNM	8.46	119.40	144.94	155.33	484.25	645.03	1428.98	1753.28
HC-PIN	2.78	14.68	55.02	13.99	34.52	663.50	234.69	372.31

6 Protein Complex Discovery

We validated our results by comparing the communities detected by FAC-PIN with a list of protein complexes obtained from the MIPS database, which we consider as a *gold standard* data. Our validations were done only for four species which we could download corresponding complexes from MIPS. For *Baker's yeast*'s PIN, we obtained complexes from the MIPS *Comprehensive Yeast Genome Database-CYGD*[3]. For the PINs of *Street rat, House mouse, and Human*, the corresponding complexes were downloaded from the MIPS *Comprehensive Resource of Mammalian Protein Complexes- CORUM* [4]. We could not find complexes for the remaining species in due time.

We proceeded similarly to Laarhoven et al. [10] and considered only the known complexes (i.e., not those obtained by computational means) containing at least three proteins. Since FAC-PIN generates non-overlapping communities, we considered only complexes which are at the bottom of the MIPS hierarchy of complexes and subcomplexes. The unconfirmed complexes, that is those in category 550, were excluded.

The validation proceeds by determining the degree of overlap between the communities identified by FAC-PIN and the protein complexes; that is, we want to determine how effectively a community matches a known complex. We used the *overlapping score* function given in [2, 3, 10, 19]. The overlapping score, $O(C, K)$, between a community C and a known complex K is defined as:

$$O(C, K) = \frac{|C \cap K|^2}{|C| \times |K|}, \tag{6}$$

A community C is considered to match a known complex K whenever $O(C, K) \geq \tau$; where, $0 < \tau \leq 1$ is the matching threshold. We have a perfect match only when $O(C, K) = 1$. Threshold value $\tau = 0.2$ was used in [2, 3, 19] whereas [10] used $\tau = 0.25$. We used both values of τ in our complex validation. After computing the overlapping scores between all pairs (C, K) of communities and known complexes for a given PIN, we then determined the ability of FAC-PIN to correctly classify the known complexes. The reason for doing this is that a given complex K_1 may match many communities but with different degrees of overlap, while another complex K_2 may match with a single community only. Hence, we calculated the *Specificity*, the *Sensitivity*, and the F-*score*, as our measures of accuracy; they are defined as follow:

[3] `ftp://ftpmips.gsf.de/yeast/catalogues/complexcat/`

[4] `http://mips.helmholtz-muenchen.de/genre/proj/corum/`

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$Specificity = \frac{TP}{TP + FP} \tag{8}$$

$$F\text{-score} = \frac{2 \times specificity \times sensitivity}{specificity + sensitivity} \tag{9}$$

where, *TP* (true positive) is the number of the identified communities C matched by the known complexes K, *FN* (false negative) is the number of known complexes that are not matched by the communities, and *FP* (false positive) is the total number of the identified communities C minus *TP*. Table 4 shows the comparison results on the protein complexes of the *Specificity*, the *Sensitivity*, and the F-*score* of FAC-PIN, HC-PIN and CNM. The results are shown for the two values of threshold τ (discussed above) and for the modularity scoring function Q. For HC-PIN, results are shown for two values of its parameter λ ([19] showed validation results with these two values of λ).

In Table 4, we see that FAC-PIN identifies communities whose average sizes (column 8) are closer to the average sizes of the known protein complexes (column 4), whereas HC-PIN and CNM yield larger averages of cluster sizes. The consequence of this is that smaller FAC-PIN communities produce higher accuracy (*Specificity*, *Sensitivity* or F-*score*) in the great majority of cases. This is because, most of the known complexes are small, and thus the accuracy increases as the size of a complex decreases. In particular, we obtain a larger number of perfectly matched complexes to communities with FAC-PIN than with HC-PIN or CNM.

7 Conclusion

In this paper, we devised a new agglomerative clustering approach, FAC-PIN algorithm, for detecting the communities of a given PIN networks, and then compared our method with two fast hierarchical techniques discussed in literature. Our approach is based on a the use of new measure, the *relative vertex clustering value* which helps decide whether a given vertex u should be included within the cluster of another vertex v depending on how many of the neighbors of u form a triangle with u and v. Our approach is very fast since we are examining only the vertices and in an efficient manner, unlike the two compared algorithms which examine edges. Thus our method is appropriate for PINs, which in general contain more interactions than proteins. More study needs to be done and we plan to perform validations based (1) on random networks, in order to analyze the robustness of FAC-PIN, and (2) on gene ontology annotations. Comparisons with other methods which are not necessarily hierarchical will also be important. Non-agglomerative clustering methods based on the relative vertex clustering value will be investigated. Finally, we plan to validate FAC-PIN through *functional enrichment* in order to evaluate how well the identified communities match with know protein functions.

Table 4. Comparison of the *Specificity*, *Sensitivity* and F*-score FAC-PIN*, *CNM* and *HC-PIN*

Species	Number of Proteins	Number of Complexes	Average Complex Size	Threshold τ	Algorithms	Computed results					
						Number of Clusters	Average Cluster Size	Perfectly Matched K	*Sensitivity*	*Specificity*	F*-score*
Baker's yeast	1237	267	4.63	0.2	*FAC-PIN*	285	4.34	12	0.092	0.78	0.164
					CNM	300	4.12	5	0.010	0.33	0.013
					HC-PIN($\lambda = 0.5$)	153	8.08	5	0.090	0.69	0.159
					HC-PIN($\lambda = 1.0$)	111	11.14	3	0.010	0.51	0.019
				0.25	*FAC-PIN*	285	4.34	12	0.090	0.82	0.162
					CNM	300	4.12	5	0.010	0.33	0.013
					HC-PIN($\lambda = 0.5$)	153	8.08	5	0.090	0.55	0.154
					HC-PIN($\lambda = 1.0$)	111	11.14	3	0.008	0.50	0.015
Human	2555	575	4.44	0.2	*FAC-PIN*	607	4.21	8	0.005	0.74	0.010
					CNM	639	3.99	5	0.004	0.40	0.007
					HC-PIN($\lambda = 0.5$)	129	19.80	3	0.005	0.39	0.009
					HC-PIN($\lambda = 1.0$)	119	21.47	3	0.004	0.44	0.007
				0.25	*FAC-PIN*	607	4.21	8	0.005	0.74	0.010
					CNM	639	3.99	5	0.004	0.31	0.008
					HC-PIN($\lambda = 0.5$)	129	19.80	3	0.005	0.39	0.009
					HC-PIN($\lambda = 1.0$)	119	21.47	3	0.004	0.44	0.007
Street rat	557	328	1.69	0.2	*FAC-PIN*	389	1.42	7	0.250	0.43	0.316
					CNM	475	1.17	3	0.109	0.36	0.248
					HC-PIN($\lambda = 0.5$)	117	4.76	1	0.160	0.33	0.214
					HC-PIN($\lambda = 1.0$)	117	4.76	1	0.160	0.33	0.214
				0.25	*FAC-PIN*	389	1.42	7	0.170	0.29	0.214
					CNM	475	1.17	2	0.150	0.27	0.192
					HC-PIN($\lambda = 0.5$)	117	4.76	1	0.110	0.22	0.143
					HC-PIN($\lambda = 1.0$)	117	4.76	1	0.110	0.22	0.143
House mouse	935	460	2.03	0.2	*FAC-PIN*	568	1.64	13	0.230	0.59	0.327
					CNM	605	1.54	6	0.120	0.56	0.198
					HC-PIN($\lambda = 0.5$)	241	3.87	3	0.180	0.48	0.265
					HC-PIN($\lambda = 1.0$)	151	6.19	3	0.110	0.50	0.182
				0.25	*FAC-PIN*	568	1.64	13	0.212	0.55	0.306
					CNM	605	1.54	6	0.120	0.56	0.198
					HC-PIN($\lambda = 0.5$)	241	3.87	3	0.153	0.41	0.222
					HC-PIN($\lambda = 1.0$)	151	6.19	3	0.110	0.50	0.182

References

1. Altaf-Ul-Amin, M.: Development and Implementation of an Algorithm for Detection of Protein Complexes in Large Interaction Networks. BMC Bioinformatics 7(207) (2006)
2. Bader, G.D., Hogue, C.W.: An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks. BMC Bioinformatics 7(2) (2003)
3. Chua, H.N., Ning, K., SUng, W.-K., Leong, H.W., Wong, L.: Using Indirect Protein-Protein Interaction for Protein Complex Prediction. Journal of Bioinformatics and Computational Biology 6(3), 435–466 (2008)
4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Nature 453(7191) (2005)
5. Fortunato, S.: Community detection in graphs. Elsevier Physics Reports 486, 75–174 (2010)
6. Friedel, C., Zimmer, R.: Inferring Topology from Clustering Coefficients in Protein-Protein Interaction Networks. BMC Bioinformatics 7(519) (2006)

7. Girvan, M., Newman, M.E.: Community Structure in Social and Biological Networks. Proceedings of Natural Academy of Science USA 99, 7821–7826 (2002)
8. Hartuv, E., Shamir, R.: A Clustering Algorithm Based on Graph Connectivity. Information Processing Letters 76(4-6), 175–181 (2000)
9. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature 402, C47–C52 (1999)
10. van Laarhoven, T., Marchiori, E.: Robust Community Detection Methods with Resolution Parameter for Complex Detection in Protein Protein Interaction Networks. In: Shibuya, T., Kashima, H., Sese, J., Ahmad, S. (eds.) PRIB 2012. LNCS (LNBI), vol. 7632, pp. 1–13. Springer, Heidelberg (2012)
11. Li, X.L., Tan, S., Foo, C., Ng, S.: Interaction Graph Mining for Protein Complexes Using Local Clique Merging. Genome Informatics 16, 260–269 (2006)
12. Li, M., Wang, J.X., Chen, J.E.: A Fast Agglomerative Algorithm for Mining Functional Modules in Protein Interaction Networks. In: Proceedings of First International Conference on BioMedical Engineering and Informatics (BMEI), pp. 3–7 (2008)
13. Luo, F.: Modular Organization of Protein Interaction Networks. BMC Bioinformatics 23(2), 207–214 (2007)
14. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review 69(066133) (2003)
15. Pei, P., Zhang, A.: A 'Seed-Refine' Algorithm for Detecting Protein Complexes from Protein Interaction Data. IEEE Transcations of Nanobioscience 6(1), 43–50 (2007)
16. Radicchi, F., Castellano, C., Cecconi, F.: Defining and Identifying Communities in Networks. Proceedings of Natural Academy of Sciences USA 101(9), 2658–2663 (2004)
17. Spirin, V., Mirny, L.A.: Protein Complexes and Functional Modules in Molecular Networks. Proceedings of Natural Academy of Science USA 100(21), 12123–12128 (2007)
18. Wang, R.-S., Zhang, S., Wang, Y., Zhang, X.-S., Chen, L.: Clustering complex networks and biological networks by non-negative matrix factorization with various similarity measures. Elsevier Neurocomputing 72 (2008)
19. Wang, J., Li, M., Chen, J., Pan, Y.: A Fast Hierarchical Clustering Algorithm for Functional Modules Discovery in Protein Interaction Networks. IEEE/ACM Transaction on Computational Biology and Bioinformatics 8(3) (2011)
20. Yook, S., Olvai, Z., Barabsi, A.L.: Functional and Topological Characterization of Protein Interaction Networks. Protenomics 4, 928–942 (2004)

Inferring Gene Regulatory Networks from Time-Series Expressions Using Random Forests Ensemble

D.A.K. Maduranga[1], Jie Zheng[1,2], Piyushkumar A. Mundra[1], and Jagath C. Rajapakse[1,3,4]

[1] Bioinformatics Research Center, School of Computer Engineering, Nanyang Technological University, Singapore 639798
[2] Genome Institute of Singapore, Biopolis Street, Singapore 138672
[3] Singapore-MIT Alliance, Singapore
[4] Department of Biological Engineering, Massachusetts Institute of Technology, USA
asjagath@ntu.edu.sg

Abstract. Reconstructing gene regulatory network (GRN) from time-series expression data has become increasingly popular since time course data contain temporal information about gene regulation. A typical microarray gene expression data contain expressions of thousands of genes but the number of time samples is usually very small. Therefore, inferring a GRN from such a high-dimensional expression data poses a major challenge. This paper proposes a tree based ensemble of random forests in a multivariate auto-regression framework to tackle this problem. The efficacy of the proposed approach is demonstrated on synthetic time-series datasets and Saccharomyces cerevisiae (Yeast) microarray gene expression data with 9-genes. The performance is comparable or better than GRN generated using dynamic Bayesian networks and ordinary differential equations (ODE) model.

Keywords: Gene regulatory networks, time-series gene expression data, gene regulation, Random forests, multivariate auto-regression, regression trees.

1 Introduction

A set of genes, transcription factors (regulators), mRNAs, and gene products (protein) interact among themselves to control almost all biological activities and form a gene regulatory network (GRN). Therefore, reverse engineering of GRN from gene expression data becomes an important problem. Reconstruction of regulatory networks plays a vital role in understanding of complexity, functionality and pathways of the biological systems and plays a crucial role in developing novel drugs for disease. With recent advancements of microarray technology and next generation sequencing, a vast amount of expression data has been produced. Thereafter, developments of novel computational models to infer the GRN from gene expression measurements have been more feasible.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 13–22, 2013.

Microarray technology enables us to gather both steady-state and time series gene expression data. Gene regulatory interactions among genes are not instantaneous, but they are dynamic events which occur throughout a period of time [1]. Therefore, time-series expression data are vital in studying the dynamics of the underlying biological systems. A typical time series data contains only a few time samples compared to the number of genes, and hence, inference of regulatory interaction of large number of genes from a few time points is one of the biggest challenges faced by computational biologists.

Several computational techniques have been proposed to infer GRN by using time course gene expression data. Boolean networks are the simplest and earliest models of gene networks [2,3]. Some of biological characteristics of actual GRN are illustrated by the Boolean network models [4]. On the other hand, ordinary differential equations (ODE) [5] are able to describe dynamic changes of the regulatory network and capture complex regulatory dependencies among the expression data. However, their major disadvantage is having a high-dimensional parameter space. Therefore, they require a large amount of experimental data to infer the accurate regulatory network. Dynamic Bayesian networks (DBN) based models are also popular in reconstructing GRN as they are capable of learning causal interactions among the temporal gene expressions [1],[6],[7]. Another approach is the usage of information theoretical measures such as mutual information (MI) to model the time course expression data. TimeDelay-ARACNE [8] is one of the recently proposed algorithms using MI among gene expressions. Also, several linear multivariate vector auto-regression (MVAR) techniques such as lasso regression, elastic net and ridge regression have been introduced in literature to infer GRN [9,10].

However, the performance of GRN inference techniques is still poor because the current approaches are unable to capture the complex regulatory interactions among the genes and many of these approaches are incapable of handling high-dimensional microarray expression data. Within this context, we propose an effective approach to infer GRN from time-course expression data with ensemble of random forest. Random forest method has become popular in handling high-dimensional problems [11], [12], [13], [14]. Huynh-Thu et al initially applied random forests technique to build GRN [15]. Their proposed method, namely GENIE3, showed the significant improvement in accuracy of GRN inference and it was the best performer in the DREAM4 *In Silico* Multifactorial challenge [15]. However, experiments were only performed with steady-state gene expression data(static data). Also the structure of the GRN was not built, but only provided the ranking of gene regulatory links. On the other hand, sparse linear regression based MVAR approaches has inherent limitations in modeling non-linear regulations. In this paper, to tackle the limitation of these previous approaches, we develop a random forests based MVAR approach to infer a GRN from time-series gene expression data. Using variable importance criterion derived from training random forest model and subsequently using adjusted R^2, a structure of GRN is obtained using time-series gene expression data.

The rest of the paper is organized in three sections. First, Section 2 describes the inference of GRN from time-course expression data using the tree based ensemble method of Random forests. Section 3 provides details on both synthetic and real datasets, performance metrics used in the evaluation, present the results and time complexity of the proposed approach. Finally, Section 4 concludes the paper with a discussion on obtained results along with future research directions.

2 Method

Let $(x_t^j)_{j=1}^q$ be a vector containing the gene expressions of q genes at the tth time point. Let x_t^{-j} is a vector containing gene expressions at time t of all the genes except gene j. By assuming that the expression level of given gene (j) at next time point $(t+1)$ is a function (g_j) of the expression values of other genes at current time (t), we can write

$$x_{t+1}^j = g_j(x_t^{-j}) + \epsilon_t, \forall t \tag{1}$$

where ϵ_t denotes the random noise. The static version of GRN inference with random forest assumes that the expression value of each gene depends on expression values of other genes for a given experiment(k) [15]:

$$x_k^j = f_j(x_k^{-j}) + \epsilon_k, \forall k \tag{2}$$

where x_k^{-j} is a vector containing all static gene expression data except expression data of gene j in the k^{th} experiment. The network inference procedure first decomposes the problem of recovering network structure of q genes into q different sub-problems. The j^{th} sub-problem is equivalent to finding regulators for j^{th} gene. Each sub problem has its own learning sample (LS_T^j) which is consists of input-output pairs for gene, $LS_T^j = (x_t^{-j}, x_{t+1}^j)_{t=1}^{T-1}$. Here, T denotes the total number of time points in the time series. Each sub-problem can be solved by finding an optimal function for g_j that minimizes the square error loss between the actual expression level and the predicted expression level by the function as follows:

$$\sum_{t=1}^{T-1} (x_{t+1}^j - g_j(x_t^{-j}))^2 \tag{3}$$

Each of these sub-problems can be categorized as supervised regression problem [15]. Regression problem which is defined by Eq. (3) can be solved by constructing tree models such as regression trees [16]. Accuracy of the single tree is further improved by ensemble methods where prediction outcomes of several individual trees are merged. Ensemble methods provide a combine prediction by considering all individual predictions in the ensemble. Therefore, the tree based ensemble method of random forest [11] is suitable for solving above problem because it can handle high dimensional expression data [13], and is capable of learning non-linear relationships as well as dealing with interacting features [15]. So, each sub-problem is solved by building an ensemble consists of regression trees using

random forest method. On the other hand, proposed method can be identified as another way of solving sparse autoregressive model where function g_j is assumed to be a linear function of the regression coefficients (β) [9,10].

First step of the random forest is generation of bootstrap samples from the initial input data. Then, each tree is constructed by using these samples. But tree building process is little bit different than the normal process because at each node, N numbers of predictors are randomly selected from the bootstrap sample to determine the optimal split for the node. The value of N is the tuning parameter because it determines the level of randomization of the trees. All the trees of an ensemble are built by applying above process.

Function g_j is learned from the learning sample LS_T^j using random forest ensemble. Following [15], weight for having a regulatory link from any gene i to j ($w_{i,j}$) are obtained by computing variable importance measure using following equation:

$$I = \#S.Var(S) - \#S_t.Var(S_t) - \#S_f.Var(S_f) \tag{4}$$

where S indicates the input data sample that reach the node, # shows the cardinality of data sample, S_f and S_t shows the subset of samples out of input data sample (S) that the test is false and true, respectively. For each subset of samples (S_f and S_t), the variance of the target variable is indicate by $Var(.)$. Variable importance measure provides an indication about the relevance of an input variable for the prediction of the output. After that, regulatory links are ranked based on their weights for each learning sample. Regulatory links that have higher weights are more likely to be actual regulatory interactions. Therefore, we apply adjusted coefficient of determination (Adjusted R^2) which is given by Eq. (5) to each sub problem to determine the actual regulators.

$$\text{Adjusted coefficient of determination } = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{5}$$

where n denotes the size of the learning sample, p is the number of regressors in the model and R^2 is the coefficient of determination. In our case, n equals to q. An important property of adjusted R^2 is that when a regression variable is added into the model , adjusted R^2 increases if added variable improves the prediction ability of the model, otherwise the value of adjusted R^2 decreases [17]. So, for each sub-problem, we add regulators into the model from highest weight to lower one and each time the value of adjusted R^2 is computed. If added regulator increases adjusted R^2, we consider it as an actual regulator. We continue adding more regressor until adjusted R^2 starts to decrease. This way, we determine the actual regulators for each sub problem.

3 Experiments and Results

Several synthetic gene expression datasets were generated and used to evaluate the performance of the proposed method. Many gene regulatory network inference studies with synthetic datasets were done using scale-free synthetic networks that were obtained using Barabasi-Albert model [18]. But in this study,

we used GeneNetWeaver (GNW) [19] software package to extract sub-networks from global Escherichia coli (E. Coli) network. Sub-networks of having 10, 30, 50 and 100 genes were extracted from E. Coli network. Topology or the structure of the gene regulatory network which has q number of genes is depicted by the connectivity matrix $M = \{M_{ij}\}_{q \times q}$ where $M_{ij} = 1$ for the presence of connection between gene i and j, and $M_{ij} = 0$ for the absence. These network topologies were used in the section 3.1 to generate synthetic gene expression data. Other than synthetic data, real time-course gene expression dataset were also used to evaluate the performance of the proposed method.

3.1 Synthetic Expression Data Generation

First-order multivariate vector autoregressive model (MVAR) [10],[9] is used to generate synthetic time-series gene expression data. Sub-networks extracted from GNW were used as network topologies in MVAR model to simulate the expression data. Gene expression at time t were obtained by using the first order MVAR model as follows:

$$x_t = x_{t-1} \times M_{weight} + \epsilon_t \tag{6}$$

where $x_t = (x_t^j)_{j=1}^q$ indicates the expressions of q number of genes at time t and ϵ_t denotes the added Gaussian random noise to the gene expression at time t. Matrix M_{weight} is obtained by assigning weights randomly to all the connection (where $M_{ij} = 1$) in the connectivity matrix M. These weights were assigned by getting the values from uniform distribution on the interval [-1,-0.6] and [0.6, 1]. Two intervals are chosen to maintain the amount of negative and positive weights nearly equal [10]. Gene expression vector at $t = 0$ ($x_{t=0}$) is initialized by obtaining the samples from the uniform distribution on the interval [0, 1] and subsequent time points are simulated using Eq. (6). For each network topology, three synthetic datasets which have 10, 30 and 50 time points were generated. For each combination of genes and time points, 50 different datasets were generated.

3.2 Real Dataset

Performance evaluation of GRN inference techniques on real gene expression data is more difficult because of lack of experimentally verified ground truth gene networks. In this study, we choose an experimentally identified gene regulatory network which is related to yeast Saccharomyces cerevisiae cell cycle [20]. This real gene regulatory network is depicted in figure 1(a) and consists of 9 genes (Fkh2, Swi4, Swi5, Swi6, Ndd1, Ace2, Cln3, Mbp1, Mcm1). Real time-series gene expression data were obtained from Spellman [21] dataset. Spellman dataset contains expression data of yeast cell cycle regulation. We selected time-course gene expression data from cdc28 cell cycle arrest which consists of 17 time points.

3.3 Performance

We generated synthetic datasets using MVAR model with the network topologies which were extracted from GNW software. Therefore, true structure of

extracted gene regulatory networks is known. Also in the real data, true structure is available since we used an experimentally verified regulatory network. Hence, we compared GRN which was inferred by the proposed random forest based approach with the ground truth network to evaluate the performance. In synthetic data, there were 50 time series datasets for each combination of genes and time points, resulting in 50 inferred GRNs. Number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) were computed for each predicted network by comparing predicted network with ground truth network. Then performance measures such as precision[1], recall[2], accuracy[3] and F-measure[4] were calculated.

For both synthetic and real dataset, an ensemble of 1000 trees was constructed. The most important parameter of this method is the number of predictors which were selected randomly to find the best split in each node. This parameter was set to $\sqrt{q}$, where q denotes the number of genes in the network. Table 1 shows the performance of the proposed method with synthetic data. In table 1, the mean and the standard deviation of each performance metric over 50 times simulation are shown. The effectiveness of the proposed method is also shown over real gene-expression data. In order to compare with existing techniques, three techniques, namely the random forest static version, dynamic Bayesian networks with Markov chain Monte Carlo (Dbmcmc software package) [1],[22] and the ordinary differential equation based model (TSNI software package)[23] were applied to the same real dataset. All the packages were used with the default settings according to their user manuals. Table 2 shows the performance measures on real data. In figure 1(b), 1(c), 1(d) and 1(e), we illustrate the gene network structures inferred from real data by the proposed method, random forests static version, ODE and DBN methods respectively. In figure 1, we used solid line to represent the true positive (TP) and dash line to represent the false negatives (FN). False positives are not shown in figure 1, though they were considered in calculating performance metrics in table 2.

3.4 Time Complexity

Random forest algorithm has time complexity of $O(Tree_{Total} * N * TlogT)$ [15], where $Tree_{Total}$ represents the number of trees in the ensemble, T denotes the number of time point in the learning sample and N denotes the number of genes that are randomly chosen at each node during construction of each tree. The proposed approach divides the infer of GRN with q number of gene into q number of sub problems. For each sub problem, we computed a value of adjusted R^2 for all regulators from highest weight to lower one. Therefore, time complexity of each sub problem became $O(q*Tree_{Total}*N*TlogT)$. Since there are altogether

[1] $Precision = \frac{TP}{FP+TP}$.
[2] $Recall = \frac{TP}{FN+TP}$.
[3] $Accuracy = \frac{TP+TN}{TP+TN+FN+TP}$.
[4] $F - measure = 2 \times \frac{Precision \times Recall}{Precision+Recall}$.

Table 1. The performance of the proposed method on synthetic data

Number of genes	Number of time points	Precision	Recall	Accuracy	F-measure
10	10	0.40 ± 0.08	0.50 ± 0.10	0.80 ± 0.03	0.45 ± 0.09
	30	0.58 ± 0.07	0.76 ± 0.09	0.88 ± 0.03	0.66 ± 0.08
	50	0.65 ± 0.07	0.86 ± 0.08	0.90 ± 0.04	0.74 ± 0.07
30	10	0.17 ± 0.03	0.43 ± 0.07	0.86 ± 0.01	0.25 ± 0.04
	30	0.32 ± 0.02	0.80 ± 0.05	0.90 ± 0.01	0.46 ± 0.03
	50	0.36 ± 0.05	0.90 ± 0.04	0.91 ± 0.00	0.52 ± 0.02
50	10	0.14 ± 0.02	0.39 ± 0.04	0.87 ± 0.00	0.21 ± 0.02
	30	0.24 ± 0.02	0.66 ± 0.07	0.89 ± 0.01	0.35 ± 0.04
	50	0.28 ± 0.02	0.78 ± 0.05	0.90 ± 0.00	0.42 ± 0.03
100	10	0.11 ± 0.01	0.30 ± 0.04	0.90 ± 0.00	0.13 ± 0.02
	30	0.14 ± 0.03	0.53 ± 0.03	0.91 ± 0.01	0.22 ± 0.04
	50	0.19 ± 0.02	0.71 ± 0.01	0.93 ± 0.01	0.30 ± 0.02

Table 2. The Performance measures on real data

Method	Precision	Recall	Accuracy	F-measure
Random forests static version	0.25	0.29	0.66	0.27
Random forests dynamic version(proposed method)	0.33	0.40	0.70	0.36
TNSI	0.28	0.29	0.69	0.29
DBN-MCMC	0.26	0.38	0.70	0.30

q number of sub problems, proposed approach has time complexity of $O(q^2 * Tree_{Total} * N * TlogT)$.

4 Discussion

Building GRN from time-series gene expression data is very important since they contain temporal information about the underline regulatory interactions among genes. In this paper, we have proposed an approach to build GRN using ensemble of random forest. The proposed approach first divides the recovering of regulatory network which is having q genes in to q different supervised regression problems. Then each of these sub problems is solved by applying random forest ensemble method. There are two main contributions of this paper. They are, 1) extend the work of [15] to infer GRN from time-series gene expression data by developing random forest based MVAR approach and 2) introduce adjusted coefficient of determination to construct the structure of GRN.

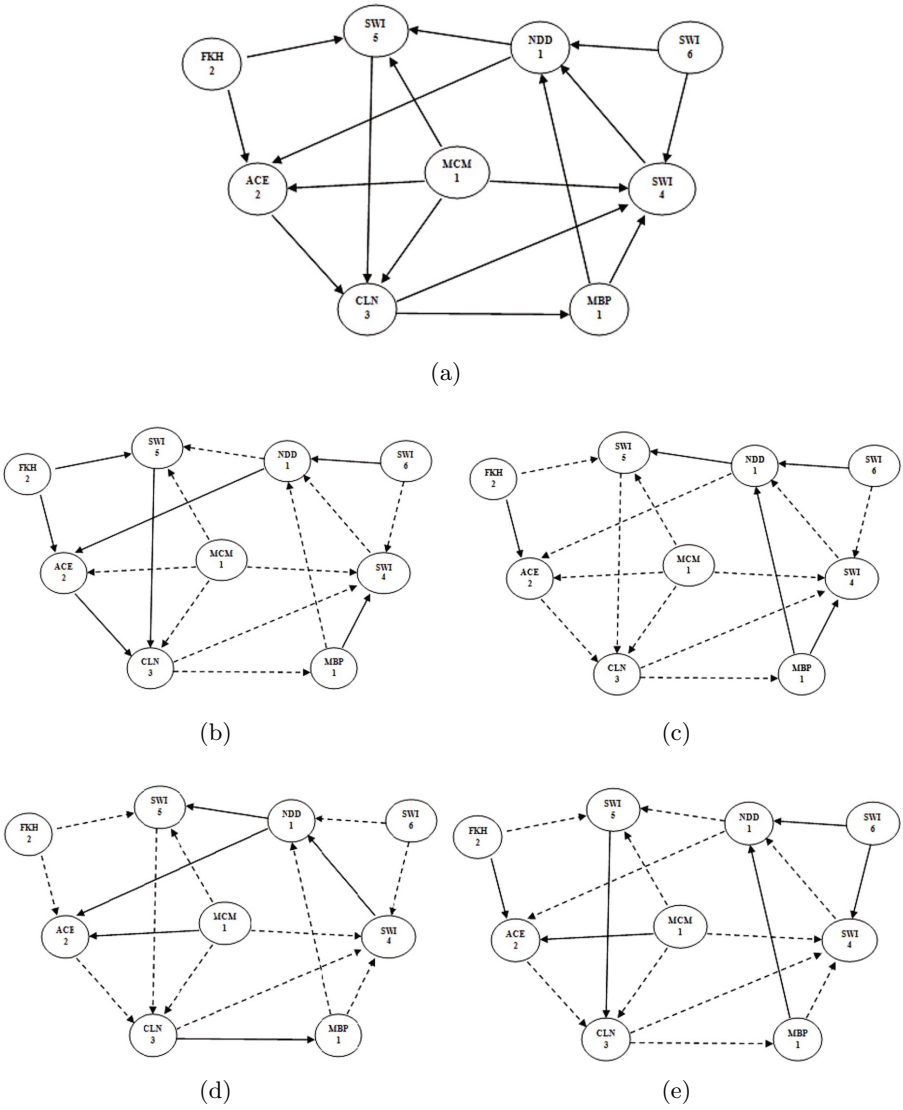


Fig. 1. The GRN identified in Yeast cell cycle and predicted network by various methods. a) is the real GRN related to yeast cell cycle [20]; b) is the predicted network by proposed approach; c) is the predicted network by Random forests static version; d) is the predicted network by TSNI; e) is the predicted network by Dbmcmc

The results on synthetic data show that all performance metrics are improved with increase in number of time points and are deteriorated with increase in number of genes. The decrease in the performance of inferred network is due to the inference of large number of false positives than false negatives. Further,

the effect of false negatives is corrected quickly than false positive effect with the increased in number of time points in the proposed method. It can also be seen that all the predicted gene networks have more than 80% of accuracy. Figure 1(b) shows the predicted GRN on the real data by the proposed random forest based approach and it is apparent that many true regulatory connections have been identified. As shown in table 2, the proposed method shows better performance on the real data compared to the Random forests static version, DBN with MCMC and ODE method.

Experiments results on both synthetic data and real expression data on a 9-gene network in yeast show the effectiveness of proposed approach. On the other hand, the proposed approach could be improved further. For example, in this study, we assumed that only gene expressions affect the gene regulation. But gene regulation also depends on other mechanisms such as histone modification and transcription factor bindings. Chen et al [24] recently showed that accuracy of DBN can be improved by integrating epigenetic data in to GRN inference. As a future work, similar approaches of data integration with random forest could improve the performance. The proposed approach divides the inference of GRN with q gene into q number of sub-problems. Since each sub-problem is independent of each other, another future work would be to parallelize all these sub-problems to reduce the computation time. Last but not least, similar to [25], the proposed method could be extended to model the time-delayed gene regulations.

Acknowledgments. This work is supported by a AcRF Tier 2 grant MOE2010-T2-1-056 (ARC 9/10), Ministry of Education, Singapore.

References

1. Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. Bioinformatics 19(17), 2271–2282 (2003)
2. Bornholdt, S.: Boolean network models of cellular regulation: prospects and limitations. Journal of the Royal Society Interface 5(suppl. 1), S85–S94 (2008)
3. Li, P., Zhang, C., Perkins, E.J., Gong, P., Deng, Y.: Comparison of probabilistic boolean network and dynamic bayesian network approaches for inferring gene regulatory networks. BMC Bioinformatics 8(suppl. 7), S13 (2007)
4. Filkov, V.: Identifying gene regulatory networks from gene expression data. Handbook of Computational Molecular Biology, 27-1 (2005)
5. Liu, B., Thiagarajan, P.S., Hsu, D.: Probabilistic approximations of signaling pathway dynamics. In: Degano, P., Gorrieri, R. (eds.) CMSB 2009. LNCS (LNBI), vol. 5688, pp. 251–265. Springer, Heidelberg (2009)
6. Kim, S.Y., Imoto, S., Miyano, S.: Inferring gene networks from time series microarray data using dynamic bayesian networks. Briefings in Bioinformatics 4(3), 228–235 (2003)
7. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 139–147. Morgan Kaufmann Publishers Inc. (1998)

8. Zoppoli, P., Morganella, S., Ceccarelli, M.: TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. Bmc Bioinformatics 11(1), 154 (2010)
9. Fujita, A., Sato, J., Garay-Malpartida, H., Yamaguchi, R., Miyano, S., Sogayar, M., Ferreira, C.: Modeling gene expression regulatory networks with the sparse vector autoregressive model. BMC Systems Biology 1, 39 (2007)
10. Rajapakse, J.C., Mundra, P.A.: Stability of building gene regulatory networks with sparse autoregressive models. BMC Bioinformatics 12(suppl. 13), S17 (2011)
11. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
12. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. BMC Bioinformatics 9(1), 307 (2008)
13. Cutler, A., Cutler, D.R., Stevens, J.R.: Tree-based methods. High-Dimensional Data Analysis in Cancer Research, 1–19 (2009)
14. Boulesteix, A.L., Janitza, S., Kruppa, J., König, I.R.: Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics (2012)
15. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PLoS One 5(9), e12776 (2010)
16. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. Chapman & Hall/CRC (1984)
17. Pagano, M., Gauvreau, K., Pagano, M.: Principles of biostatistics. Duxbury Pacific Groveˆ eCA CA (2000)
18. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
19. Marbach, D., Schaffter, T., Mattiussi, C., Floreano, D.: Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Journal of Computational Biology 16(2), 229–239 (2009)
20. Simon, I., Barnett, J., Hannett, N., Harbison, C.T., Rinaldi, N.J., Volkert, T.L., Wyrick, J.J., Zeitlinger, J., Gifford, D.K., Jaakkola, T.S., et al.: Serial regulation of transcriptional regulators in the yeast cell cycle. Cell 106(6), 697–708 (2001)
21. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell 9(12), 3273–3297 (1998)
22. Husmeier, D.: Inferring dynamic bayesian networks with mcmc (2003), `http://www.bioss.ac.uk/~dirk/software/DBmcmc/index.html`
23. Bansal, M., Della Gatta, G., Di Bernardo, D.: Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. Bioinformatics 22(7), 815–822 (2006)
24. Haifen, C., Maduranga, D., Mundra, P., Zheng, J.: Integrating epigenetic prior in dynamic bayesian network for gene regulatory network inference. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (accepted, 2013)
25. Mundra, P., Niranjan, M., Welsch, R., Zheng, J., Rajapakse, J.: Inferring time-delayed gene regulatory networks using cross-correlation and sparse regression. In: 9th International Symposium on Bioinformatics Research and Applications (accepted, 2013)

Local Topological Signatures for Network-Based Prediction of Biological Function

Wynand Winterbach[1,2], Piet Van Mieghem[1], Marcel J.T. Reinders[2,3,4], Huijuan Wang[1], and Dick de Ridder[2,3,4]

[1] Network Architectures and Services Group
[2] Delft Bioinformatics Lab,
Department of Intelligent Systems, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, P.O. Box 5031, 2600 GA Delft
[3] Netherlands Bioinformatics Center
6500 HB Nijmegen, The Netherlands
[4] Kluyver Centre for Genomics of Industrial Fermentation
2600 GA Delft, The Netherlands
w.winterbach@tudelft.nl

Abstract. In biology, similarity in structure or sequence between molecules is often used as evidence of functional similarity. In protein interaction networks, structural similarity of nodes (i.e., proteins) is often captured by comparing node signatures (vectors of topological properties of neighborhoods surrounding the nodes).

In this paper, we ask how well such topological signatures predict protein function, using protein interaction networks of the organism *Saccharomyces cerevisiae*. To this end, we compare two node signatures from the literature – the graphlet degree vector and a signature based on the graph spectrum – and our own simple node signature based on basic topological properties.

We find the connection between topology and protein function to be weak but statistically significant. Surprisingly, our node signature, despite its simplicity, performs on par with the other more sophisticated node signatures. In fact, we show that just two metrics, the link count and transitivity, are enough to classify protein function at a level on par with the other signatures suggesting that detailed topological characteristics are unlikely to aid in protein function prediction based on protein interaction networks.

1 Introduction

To what extent does structure determine function in biology? Evolutionary principles have shown function and structure to be well correlated in genes with common evolutionary ancestors, allowing biologists to infer functions of proteins or genes based on their sequence *homology* (i.e., similarity) with other proteins or genes. With the arrival of network biology [1], homology was extended to take not only sequence similarity into account but also similarity of molecular

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 23–34, 2013.

interactions. These interactions can be either direct (physical) or indirect (functional). In other words, the manner in which a protein (or gene) is connected to other proteins in interaction networks matters. These other connecting proteins can be chosen in many ways, although the most common approach is to consider a network neighborhood centered around a protein in question, including all proteins and links within a fixed number of hops. Structural similarity of network neighborhoods is determined by comparing their *topological* properties. Typically, these properties are represented as a vector, known as a *topological signature*.

Topological signature similarity has been used as a measure of functional similarity between proteins in several algorithms aimed at the discovery of homology relations between proteins [2–4]. Although topological similarity and amino acid sequence similarity are typically both used to determine homology [2, 4], some of these algorithms perform well using only topological similarity [3, 4]. Researchers have also used topological similarity to predict relations other than homology, in effect assuming that structural similarity implies similarity of biological traits in proteins not necessarily related by evolution. Involvement in cancer (a phenotype) was found to be encoded in topological similarity [5] and even general protein function appears to be encoded in topology [6]. Given this predictive quality, the key question is thus: how exactly does local topology reflect function, and what signatures best capture local topology?

In this paper, we set out to answer these questions in a specific context, i.e. the prediction of protein function by means of node signatures in various protein interaction networks of the organism *Saccharomyces cerevisiae*. Topological signatures in the literature capture a lot of topological detail; in this paper we investigate the extent to which this detail improves protein function prediction (if at all). To this end, we study two such signatures – the graphlet signature of Milenković and Pržulj [6] and a signature based on the normalized Laplacian spectrum of a network [4] – as well as a simple node signature of our design. Predictive power of the signatures is determined by how well they discriminate between proteins with a given biological function and those without the function. To this end we use support vector machines, treating topological signatures as feature vectors and biological labels as classifier labels. Note that our aim is not the construction of an optimal protein function classifier, as for that purpose one would include many other types of data; rather, we use prediction accuracy as a measure to explore the relation between local topology and function.

2 Methods

2.1 Topological Signatures

In the remainder of the text, G refers to a network (usually an interaction network), n to an arbitrary node of G and N the number of nodes in G. A k-neighborhood G_n^k of a node n is an induced subnetwork of G on the set of nodes encompassing n and all nodes within k hops of n (a subnetwork is induced when two nodes in the subnetwork are connected by a link if, and only if they are

(a) G_n^1. (b) G_n^2.

Fig. 1. Two neighborhoods of n: (a) G_n^1 and (b) G_n^2

connected in G). The subnetwork G_n^1 spanned by the gray nodes and bold links in Figure 1(a) is a 1-neighborhood of n, whilst the subnetwork G_n^2 spanned by the gray nodes and bold lines in Figure 1(b) is a 2-neighborhood of n.

Graphlet Signature: Graphlets are small, connected, induced subnetworks, as illustrated in Figure 2. The graphlet degree of a node n can be regarded as a generalization of its degree: the number of graphlets of a specific type $(X_1, X_2, \ldots)$ that contains n (the degree is the number of X_1 subgraphs containing n). A graphlet signature (also graphlet degree sequence [6]) generalizes the graphlet degree by including counts for all of the subnetworks in Figure 2.

To simplify exposition, we first construct a graphlet signature containing only the numbers of subnetworks X_1, X_2 and X_3 (Figure 2) that contain n. Such a signature can be represented as a vector of three integers. However, X_2 is not symmetrical, as the white node is structurally different from the two black nodes (which are interchangeable). We distinguish cases in which n takes the role of the white node from cases in which n takes the role of the black nodes. Thus, two counts for X_2 are maintained (one for each kind of node), leading to a signature vector of four integer components: one for X_1, two for X_2 and one for X_3 (vector indices are shown next to one node of each color).

The full graphlet signature is constructed by extending the construction above to the rest of the subnetworks in Figure 2. In total, the signature vector has 73 components (vector indices appear next to nodes). The largest subnetworks in Figure 2 have five nodes and therefore the graphlet signature is computed on 4-neighborhoods. The larger subnetworks in Figure 2 contain induced copies of smaller subnetworks (e.g., X_{30} contains X_9, X_3 and X_1), so that the components of the graphlet signature are not independent. Milenković and Pržulj [6] devised a weighting scheme to reduce this effect. We reweigh graphlets according to their method. Graphlet signatures were computed using code adapted from the original version of GraphCrunch [7].

Spectral Signature: We assume that the nodes in G are labeled with numbers 1 through N. The *adjacency matrix* A of G is an $N \times N$ matrix in which $a_{i,j} = 1$ if the nodes i and j are connected by a link and $a_{i,j} = 0$ otherwise. The degree matrix Δ of G is a matrix in which $a_{i,i}$ equals the degree of node i and $a_{i,j} = 0$

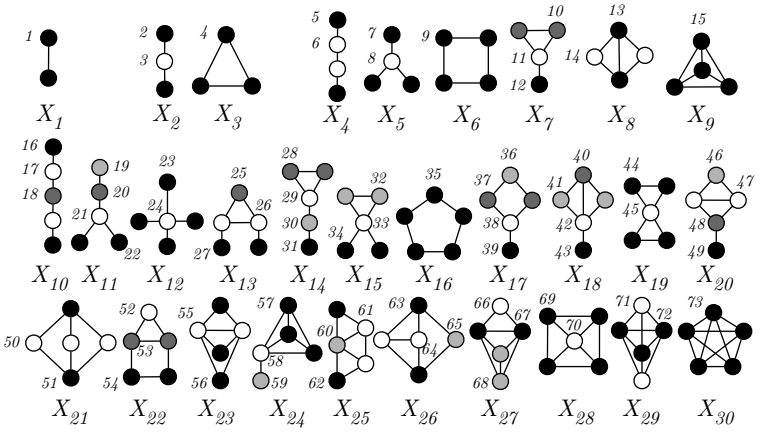


Fig. 2. All non-isomorphic undirected networks (graphlets) with up to five nodes. For a given node n in a network G, Milenković & Pržulj [6] count how many times each of these networks includes n and appears as an induced subnetwork in G in order to construct a graphlet signature for n.

if $i \neq j$. The *normalized Laplacian* is defined as $Q_{\text{norm}} = I - \Delta^{-1/2} A \Delta^{-1/2}$. The *spectrum* of Q_{norm} is its set of N eigenvalues. All eigenvalues of Q_{norm} fall within the range of $[0, 2]$.

In general, two different neighborhoods have different numbers of nodes and therefore spectra of different sizes, making spectra unsuitable as feature vectors. We derive feature vectors by computing histograms of the spectra [4]. Histograms with 20 bins are computed on the range $[0, 2]$, showing why the normalized Laplacian spectrum is preferred over the non-normalized version.

Simple Metric Signature: Our own simple metric signature serves as a baseline. It contains four very simple topological properties of neighborhoods: 1) number of nodes, 2) number of links, 3) link density and 4) transitivity (the ratio of triangles to connected node triplets).

Multi-resolution Signatures: One way to compute the spectral and simple metric signatures is to choose a fixed k and to compute the signatures on all k-neighborhoods. By focusing on fixed k, one may miss topologically distinguishing features at other "resolutions", i.e., other values of k. We construct "multi-resolution" versions of the spectral and simple metric signatures respectively by concatenating signatures of G_n^1, G_n^2 and G_n^3 for a given node n; henceforth we shall only consider these "multi-resolution" versions of the signatures. The graphlet signature is already "multi-resolution" in the sense that its component graphlets span G_n^1, G_n^2, G_n^3 and G_n^4.

A Combined Signature: Finally, we consider a signature that combines the previous signatures by simply concatenating the 1) graphlet signature, 2) the multi-resolution spectral signature and 3) the multi-resolution simple metric signature.

2.2 Datasets

Molecular Networks: All of the networks considered in this paper are protein interaction networks for the organism *Saccharomyces cerevisiae*. We have collected seven such networks, derived from four primary sources. Kim & Marcotte [8] provide two protein interaction networks, the first a high-quality literature-curated network and the second a high-throughput network. Yeastnet [9] provides several datasets with yeast protein interactions of which we downloaded the literature-curated dataset (denoted "LC" on the website) and the yeast 2-hybrid high-throughput dataset ("HC"). These two pairs of networks were selected because each pair contains a literature curated network and a high-throughput network, thereby providing insight into the impact of network quality on classification performance.

Our remaining two datasets are due to Krogan [10] and von Mering [11]. Both of these were used by Milenković & Pržulj [6] to test how well their graphlet signature approach fared in predicting protein function. We used the same two subsets of the von Mering dataset: "von Mering" contains the first 11000 protein interactions (of high-, medium- and low-confidence), whilst "von Mering core" contains all high-confidence interactions of the original dataset.

Biological Labels: Like Milenković and Pržulj [6], we used the MIPS protein annotations [12] as biological labels. MIPS annotations are hierarchical and have the form "xx.yy.zz..." where the letters denote two-digit biological categories. A protein may be annotated with multiple such annotations. The left-most category ("xx") gives the general protein function; each following two-digit category is a refinement ("yy" and "zz"). In this paper, we consider only general protein functions, of which there are 27 in the MIPS database.

2.3 Classification

Classification is performed using support vector machines (SVMs). There are numerous biological categories in the MIPS database and a protein may be annotated with any number of these categories. Since SVMs are binary classifiers, we use a one-versus-all strategy whereby we train a classifier for each biological category. Classifier performance is measured using the area under the curve (AUC) of the receiver operator curve (ROC) of a classifier. All classifier-related work was performed using Scikit-learn [13].

The radial basis function (RBF) kernel was used to train all SVMs. To reduce the impact of experimental omissions and noise, we only compute signatures on nodes whose degrees are at least 3 and that have at least one MIPS annotation.

Furthermore, to ensure the presence of enough positive instances in both testing and training sets, biological labels that appear in less than 20 nodes are not considered for classification training.

Training Regime: For each topological signature type, for each network, for each biological function, a double cross validation training loop is performed [14]. The "outer" loop is a four-fold loop in which the training set contains 75% of the dataset whilst the testing set contains 25% of the dataset. For a given network and biological function, the folds are fixed, meaning that classifiers are trained on the same training samples for all topological signatures. Classifier performance is expressed as a combination of the mean and standard deviation of the four AUC values associated with the four outer folds.

The "inner" loop is responsible for finding the classifier with the best classification performance on the training set received from the "outer" loop. SVM classifiers using the RBF kernel require two parameters: a cost C (for penalizing incorrectly classified instances) and the RBF radius γ. These are optimized by walking along a grid of parameter pairs and training a classifier for each pair. Each grid point (i.e., parameter pair) is evaluated using the average AUC of a five-fold cross-validation loop. The parameters with the best AUC score are thus considered optimal. At the start of the "inner" loop, both the training and testing sets are centered and scaled using the center and variance of the training set. The graphlet signature is reweighed after this point using the weighting scheme of Milenković and Pržulj [6] as mentioned earlier in the paper (if reweighing is applied beforehand, it would be removed by the scaling step).

As grid searches are expensive, we first perform a parameter search on a coarse grid, followed by a second search on a fine grid around the optimal parameters found in the first search. The coarse grid is given by the Cartesian product $\mathcal{C} \times \Gamma$ of costs $\mathcal{C} = \{2^{-5}, 2^{-3}, 2^{-1}, \ldots, 2^{15}\}$ and RBF radii $\Gamma = \{2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^{3}\}$. The optimal parameter pair (C, γ) discovered on $\mathcal{C} \times \Gamma$ is then used to specify a fine grid $\mathcal{C}' \times \Gamma'$ where $\mathcal{C}' = \left\{2^{\log_2 C - 2 + i/2} \mid i \in \{0, 1, \ldots 8\}\right\}$ and $\Gamma' = \left\{2^{\log_2 \gamma - 2 + i/2} \mid i \in \{0, 1, \ldots 8\}\right\}$.

3 Results and Discussion

Using the training regime described in the Methods section, we have computed, for each topological signature, for each network, for each biological function, the average classifier performance as well as its standard deviation. As this is a large amount of data, we have condensed the results into Figure 3(a) which shows, for a given topological signature and biological function, classification performance averaged over all networks, except for the high-throughput Yeastnet network. This dataset proved to be too small and gave poor, noisy classification results for all topological signatures. Figure 3(a) contains only those biological functions that appear in all the datasets. We also plotted the classification results for one high-quality dataset, the literature-curated Yeastnet dataset, in Figure 3(b).

The trends in Figure 3(a) are broadly similar in all of the networks although classification performance is generally lower than in Figure 3(b).

What stands out most from both Figure 3(a) and Figure 3(b), is that topology is, in general, a weak predictor of biological function. However, the mean AUC values are all above 0.5, showing that topology does encode a certain amount of information about biological function (the statistical significance of the mean AUC values being larger than 0.5 was tested using the t-test; in the majority of cases – and in all cases involving the biological categories "metabolism", "transcription", "protein synthesis" and "protein fate" – the associated p-values are below 0.05). The overall differences between Figure 3(a) and Figure 3(b) can be explained by differences in network quality and network size: quality affects classifier performance whilst network size affects its variance (network sizes are given in Table 1). The high-throughput networks contain the most noise and are therefore associated with worse classification performance.

At the level of biological categories both Figure 3(a) and Figure 3(b) show big differences in classification performance. The number of positive instances associated with a biological category (see Table 1) is weakly correlated with classifier performance, partly explaining the differences. Biology offers a possible explanation for the high AUC values associated with the labels "Transcription" and "Protein Synthesis": transcription and synthesis are both processes driven by permanent protein complexes rather than temporary groups of proteins (as found in many other processes). Thus, nodes with these functions tend to find themselves in densely connected clusters more often than other nodes.

Both overall classification performance, as well as performance associated with individual biological categories are dependent on the way in which biological categories are defined. Some categories are more general than others (for example,

Table 1. The number of positive instances for various combinations of network and biological function (i.e., proteins having given biological functions)

	Metabolism	Energy	Cell cycle & DNA processing	Transcription	Protein synthesis	Protein fate	Protein w. binding function	Regulation of protein function	Cellular transport	Cellular communication	Cell rescue & defense	Environment interaction	Cell fate	Development	Biogenesis	Cell type differentiation	Number of unique nodes
Kim & Marcotte, HT	271	63	377	481	130	347	399	62	207	46	123	94	80		220	128	1123
Kim & Marcotte, LC	452	84	674	676	149	655	652	157	469	134	235	221	192	35	458	288	1933
Krogan	321	81	423	483	183	378	405	70	205	61	148	115	87		277	134	1281
von Mering core	102	25	75	158	102	88	130		54		22	29	26		84	48	371
von Mering	471	120	231	382	289	295	369	49	193	39	114	99	68		222	104	1307
Yeastnet, HT	96	22	110	90		112	97	26	96	24	52	44	52		99	63	353
Yeastnet, LC	442	82	618	630	207	637	645	142	580	124	222	239	187	39	444	281	2006

"Development" includes proteins engaged in diverse functions, whereas "Transcription" is a more specific function), contributing to differences in classification performance between categories. When the categories are too general, overall classification performance suffers as classifier inputs become difficult to distinguish. We have performed experiments (data not shown) in which we used two levels of the MIPS labels (labels of the form "xx.yy" rather than just "xx", i.e., more specific categories). Two-level categories led to better classification performance in some cases (notably those associated with transcription) and worse performance in other cases. The culprit is likely a paucity of positive instances associated with many of the two-level labels.

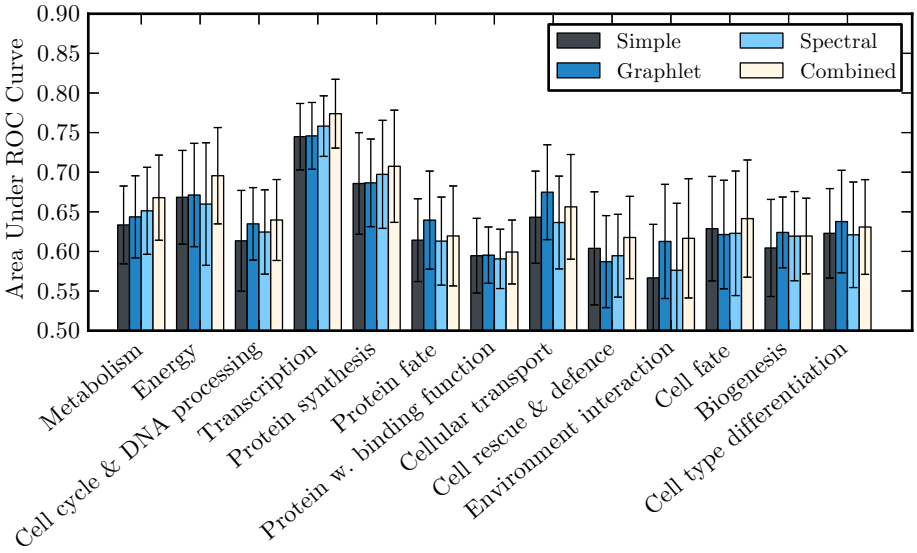
Another salient aspect of Figure 3(a) and Figure 3(b) is that the three topological signatures perform very similarly. We tested whether the AUC values of the individual signatures (i.e., not the combined signature) for each biological category were different, using a one-way ANOVA (Table 2). We consider p-values of 0.05 and below to be statistically significant and find only 10 dataset/function combinations that pass this threshold.

Although the three topological signatures lead to similar classification results, it may be possible that they nevertheless measure different (discriminative) topological characteristics. If this is true, combining the signatures should lead to improved classification performance. However, Figure 3(a) and Figure 3(b) do not support such a conclusion. Thus, in the context of our datasets and classifier, the topological signatures are not complementary.

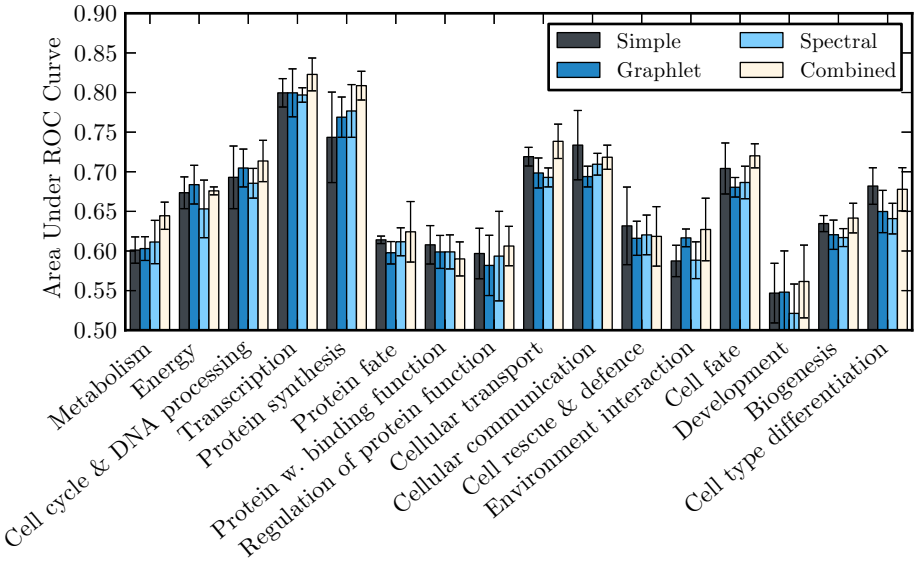
Given that the simple metric signature is competitive with the graphlet and spectral signatures, it is natural to ask whether it cannot be further

Table 2. p-values of one-way ANOVA tests applied to the AUC values of the three topological signatures (graphlet, spectral and simple) for each network and biological function combination. We consider p-values of 0.05 and below to be significant (shown in bold text).

	Metabolism	Energy	Cell cycle & DNA processing	Transcription	Protein synthesis	Protein fate	Protein w. binding function	Regulation of protein function	Cellular transport	Cellular communication	Cell rescue & defense	Environment interaction	Cell fate	Development	Biogenesis	Cell type differentiation
Kim & Marcotte, HT	.39	.95	**.04**	.17	.39	.77	.69	.14	.10	.15	.19	.23	.35		.85	.56
Kim & Marcotte, LC	.42	.91	.10	.06	**.05**	**.00**	.27	.64	**.01**	.61	.74	**.01**	.31	.76	**.05**	.70
Krogan	.94	.08	.34	.13	.26	.20	.07	.12	.47	.90	.91	.07	.43		.18	.32
von Mering core	.75	.55	.14	.08	.26	.82	.56		.79		.92	.53	.87		.53	.97
von Mering	.19	.32	.49	.12	.59	.24	.26	.14	.24	.06	.50	.43	.60		.17	**.04**
Yeastnet, HT	.44	.22	.12	.36		.68	.19	.07	.18	**.04**	.12	**.00**	.45		.69	.70
Yeastnet, LC	.80	.42	.84	.55	.60	.11	.91	.85	**.04**	.23	.93	.62	.63	**.05**	**.01**	.12



(a)



(b)

Fig. 3. Classification performance of the three topological signatures, as well as a signature that combines the three signatures. (a) Performance of our SVM classifiers averaged MIPS categories present in all datasets (excluding the high-throughput Yeastnet dataset; see text for explanation). Error bars show the standard deviation. (b) Classification performance of the three topological signatures on the literature-curated Yeastnet network [9].

simplified. We investigated all possible combinations of the four metrics (number of nodes, number of links, density and transitivity) that make up the simple metric signature, constructing 14 simpler signatures: 4 signatures using only one metric each, 6 signatures using pairs of metrics and 4 signatures using triplets of metrics. The mean classification performance of these metrics, taken over all datasets and all biological categories, is shown in Figure 4. The link count L and transitivity T are sufficient for obtaining good classification performance. The implication is that what matters in function prediction in protein interaction networks, is the number of nodes and the "clusteredness" (transitivity). Since proteins of similar function tend to form clusters, their neighborhoods overlap and therefore they share topological characteristics. Apparently, "clusteredness" signatures are unique enough to distinguish similar proteins from other proteins.

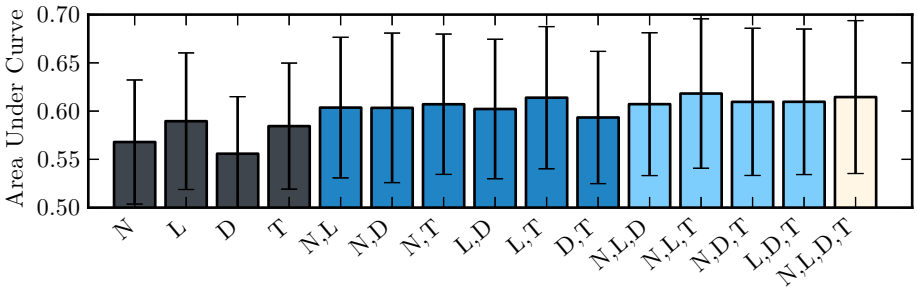


Fig. 4. Classification performance of various combinations of the features used in the simple metric signature averaged over all datasets and all functions. Here, N is the number of nodes (in a neighborhood), L is the number of links, D is the density and T is the transitivity.

4 Conclusion

At the start of this paper, we asked to what extent structure – i.e., topology – determines function in biology. We focused on the use of signatures to express topological properties of neighborhoods surrounding nodes in molecular interaction networks. Our study is motivated by the use of topological signatures as a tool for discovering similar genes or proteins (under the assumption that topological similarity implies functional similarity). We specifically studied the use of such signatures to discriminate between proteins with a given biological function and those without it, using protein interaction networks derived from *Saccharomyces cerevisiae* and support vector machines.

Current node signatures, such as the graphlet signature [6] and signatures based on spectra [4] capture very detailed topological profiles. We compared these with our own topological signature, based on very simple network metrics. For all signatures, classifier performance tended to be weak, implying that topology is, at least for *Saccharomyces cerevisiae* protein interaction networks, a weak

predictor of function. However, with the exception of one noisy protein interaction network classifiers performed better than random, showing that topology and function are linked. How much better depends on the functional category considered, with performance particularly strong for transcription and protein synthesis.

Our simple metric signature performed on par with the graphlet and spectral signatures. We also established that the signatures are not complementary for protein function prediction, as a combined signature incorporating all three signatures does not yield better accuracy. Since our simple metric signature captures less topological information than the other signatures, we conclude that fine topological detail is not very useful in the prediction of protein function. Strikingly, performance when using only the link count and transitivity, measures of "clusteredness", is as good as when using the more complex signatures. This is not simply a side-effect of dataset noise, as our simple metric signature performs equally well in the high quality networks.

Our work opens a number of paths for future research. For our conclusions to hold generally, the techniques used in this paper should be applied to other types of interaction networks (for example, co-expression networks and synthetic sick-or-lethal networks) and to networks derived from other organisms. It would be particularly interesting if link count and transitivity are found to be equally determinative in other interaction network types. Finally, it is not yet known how different "resolutions" contribute to signature performance and whether a particular resolution (i.e., k-neighborhoods of a particular k) dominates classification performance.

References

1. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabási, A.L.: Hierarchical organization of modularity in metabolic networks. Science 297(5586), 1551–1555 (2002)
2. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics 25(12), i253–i258 (2009)
3. Milenković, T., Ng, W.L.L., Hayes, W., Pržulj, N.: Optimal network alignment with graphlet degree vectors. Cancer Informatics 9, 121–137 (2010)
4. Patro, R., Kingsford, C.: Global network alignment using multiscale spectral signatures. Bioinformatics (2012)
5. Milenković, T., Memišević, V., Ganesan, A.K., Pržulj, N.: Systems-level cancer gene identification from protein interaction network topology applied to melanogenesis-related functional genomics data. Journal of The Royal Society Interface 7(44), 423–437 (2010)
6. Milenković, T., Pržulj, N.: Uncovering biological network function via graphlet degree signatures. Cancer Informatics 6, 257–273 (2008)
7. Milenkovic, T., Lai, J., Pržulj, N.: GraphCrunch: A tool for large network analyses. BMC Bioinformatics 9(1), 70 (2008)
8. Kim, W.K., Marcotte, E.M.: Age-Dependent evolution of the yeast protein interaction network suggests a limited role of gene duplication and divergence. PLoS Computatinal Biology 4(11) (November 2008)

9. McGary, K., Lee, I., Marcotte, E.: Broad network-based predictability of saccharomyces cerevisiae gene loss-of-function phenotypes. Genome Biology 8(12), R258 (2007)
10. Krogan, N.J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., Li, J., Pu, S., Datta, N., Tikuisis, A.P., Punna, T., Peregrín-Alvarez, J.M., Shales, M., Zhang, X., Davey, M., Robinson, M.D., Paccanaro, A., Bray, J.E., Sheung, A., Beattie, B., Richards, D.P., Canadien, V., Lalev, A., Mena, F., Wong, P., Starostine, A., Canete, M.M., Vlasblom, J., Wu, S., Orsi, C., Collins, S.R., Chandran, S., Haw, R., Rilstone, J.J., Gandi, K., Thompson, N.J., Musso, G., St Onge, P., Ghanny, S., Lam, M.H.Y., Butland, G., Altaf-Ul, A.M., Kanaya, S., Shilatifard, A., O'Shea, E., Weissman, J.S., Ingles, C.J., Hughes, T.R., Parkinson, J., Gerstein, M., Wodak, S.J., Emili, A., Greenblatt, J.F.: Global landscape of protein complexes in the yeast saccharomyces cerevisiae. Nature 440(7084), 637–643 (2006)
11. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., Bork, P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417(6887), 399–403 (2002)
12. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., Mewes, H.W.: The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research 32(18), 5539–5545 (2004)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
14. Wessels, L.F.A., Reinders, M.J.T., Hart, A.A.M., Veenman, C.J., Dai, H., He, Y.D., van't Veer, L.J.: A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics 21(19), 3755–3762 (2005)

Mutational Genomics for Cancer Pathway Discovery

Jeroen de Ridder[1,2,3], Jaap Kool[4,6], Anthony G. Uren[5,6], Jan Bot[1,3], Johann de Jong[2], Alistair G. Rust[7], Anton Berns[6], Maarten van Lohuizen[6], David J. Adams[7], Lodewyk Wessels[1,2,3], and Marcel J.T. Reinders[1,3]

[1] Delft Bioinformatics Lab, Delft University of Technology
[2] Bioinformatics and Statistics, Dept. Molecular Biology, Netherlands Cancer Institute
[3] Netherlands Bioinformatics Centre
[4] MSD Animal Health, Merck/Intervet B.V.
[5] MRC Clinical Sciences Centre, Imperial College Faculty of Medicine
[6] Dept. Molecular Genetics, Netherlands Cancer Institute
[7] Experimental Cancer Genetics, Wellcome Trust Sanger Institute
l.wessels@nki.nl, m.j.t.reinders@tudelft.nl

Abstract. We propose *mutational genomics* as an approach for identifying putative cancer pathways. This approach relies on expression profiling tumors that are induced by retroviral insertional mutagenesis. Akin to genetical genomics, this provides the opportunity to search for associations between tumor-initiating events (the viral insertion sites) and the consequent transcription changes, thus revealing putative regulatory interactions. An important advantage is that in mutational genomics the selective pressure exerted by the tumor growth is exploited to yield a relatively small number of loci that are likely to be causal for tumor formation. This is unlike genetical genomics which relies on the natural occurring genetic variation between samples to reveal the effects of a locus on gene expression.

We performed mutational genomics using a set of 97 lymphoma from mice presenting with splenomegaly. This identified several known as well as novel interactions, including many known targets of *Notch1* and *Gfi1*. In addition to direct one-to-one associations, many multilocus networks of association were found. This is indicative of the fact that a cell has many parallel possibilities in which it can reach a state of uncontrolled proliferation. One of the identified networks suggests that *Zmiz1* functions upstream of *Notch1*. Taken together, our results illustrate the potential of mutational genomics as a powerful approach to dissect the regulatory pathways of cancer.

1 Introduction

Cancers arise as a result of a multistep process in which genetic alterations deregulate the regulatory pathways that govern healthy cell proliferation [1]. To study this process, the use of DNA microarrays for transcriptome profiling of tumor tissue has proven useful. Success stories include, among others, finding good diagnostic and prognostic markers [2, 3], and providing insight in different tumor subtypes [4]. However, to identify the *causal* genetic alterations, transcriptome profiling is less suitable. This is because, in many cases, aberrant gene expression is a downstream effect of one or more genetic alterations elsewhere, rather than the causal event in tumor development.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 35–46, 2013.

To identify genes that are likely to have a driving role in cancer, high-throughput retroviral insertional mutagenesis (RIM) screens can be performed [5–8]. In these screens, retroviruses are used to induce insertion mutations in the genome of infected somatic cells in mice. These mutations may cause alteration in expression of genes in the vicinity of the insertion or, when inserted within a gene, alteration of the gene product. A certain proportion of these mutations are oncogenic and will result in tumor development. Consequently, the genomic location of the inserted viruses in the resulting tumors provide 'tags' for cancer genes, since regions in the genome that harbor insertions in multiple independent tumors are likely to be in the vicinity of genes that play a causal role in tumor development.

1.1 Mutational Genomics

Here, we perform genome-wide expression profiling in tumors induced by RIM. Combining expression with insertion site data provides the unique opportunity to study the relationship between the initiating events and their downstream transcriptional effect. We call this approach *mutational genomics*.

Mutational genomics bears similarity to genetical genomics, linking genotype to transcriptional state [9–11]. In the latter approach, often performed in fully genotyped recombinant inbred (RI) mouse strains, expression quantitative trait loci (eQTLs) are determined. These are defined as chromosomal regions for which the local genotype segregates the gene expression of one or more genes, and may point to putative regulators of these genes [12–14]. Similarly, mutational genomics allows the definition of, what we coin, expression quantitative mutation loci (eQMLs), i.e. chromosomal regions that are mutated in multiple independent tumors and are associated with a segregation of the expression of one or more genes. This concept is schematically illustrated in Figure 1.

A major advantage of mutational genomics is that the list of candidate target genes of the identified eQMLs is usually limited to only a few. This is because insertions act primarily on proximal genes [15] using one of a specific set of fairly well defined mechanisms [5, 7, 16]. Typical eQTLs, on the other hand, usually span large regions in the genome containing many genes as a result of linkage disequilibrium. Consequently, in mutational genomics the difficult task of finding the genes underlying the transcriptional changes is circumvented.

A second important advantage stems from the fact that mutational genomics exploits the selective pressure exerted during tumor development to yield a relatively small number of loci that are likely to be causal for tumor formation. This is unlike genetical genomics in which one has to rely on the natural occurring genetic variation between samples to reveal the effects of a locus on gene expression. As a result, eQMLs are specific for the type of tumor under study, and therefore represent important building blocks that help delineating the regulatory pathways that play a role in these tumors.

1.2 Multilocus Interactions

Cancer is a complex disease, involving the mutation and/or deregulation of multiple genes. Many of the changes that are required for tumorigenesis are a result of the collaboration between mutations of cancer genes. Moreover, for many of the mutational steps

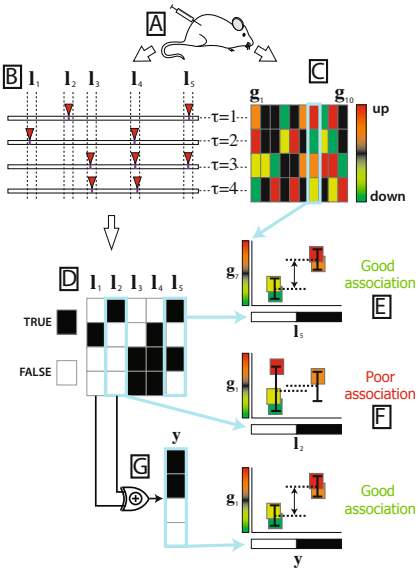


Fig. 1. Schematic overview of the data for four tumors. A) After infection with a slow transforming retrovirus, tumors are harvested. **B)** The insertion loci are retrieved by sequencing the flanking regions. The figure shows five unique insertion loci ($\mathbf{l}_1 - \mathbf{l}_5$), for four tumors ($\tau = 1, \ldots, 4$). **C)** For each tumor, gene expression profiles are determined by microarrays. The figure shows 10 genes ($\mathbf{g}_1 - \mathbf{g}_{10}$). **D)** The insertion data can be considered as a Boolean matrix. **E)** An insertion locus is said to be associated with the expression of a gene when the presence or absence of an insert segregates the gene expression in a highly expressed and lowly expressed group, as is the case for inserts in $\mathbf{l}_5$ and expression of $\mathbf{g}_7$. **F)** In some cases a single insertion locus does not suffice to explain the expression values, exemplified by the poor association between $\mathbf{l}_2$ and $\mathbf{g}_1$. **G)** Multilocus models, combining multiple loci using Boolean logic ($\mathbf{l}_1$ XOR $\mathbf{l}_2$), may be employed to explain more of the transcriptional variance.

required to transform healthy cells to cancer cells numerous alternatives exist. This is especially pertinent while analyzing mutational genomics data, since this means that many of the regulatory interactions may not be detectable as direct (marginal) associations, but rather require multivariate analysis of the data (see Figure 1G for a schematic example).

Therefore we propose to explore multilocus mapping by explicitly incorporating the possibility of alternative and collaborative pathways in the search for eQMLs. Because the presence or absence of an insertion is naturally captured by a Boolean variable, a Boolean model is used to combine insertion loci. To this end, we employ the combinatorial association logic (CAL) network inference procedure, that we recently proposed for finding multilocus interaction in a genetical genomics dataset [17]. Using CAL network inference we are able to efficiently determine the set of insertion loci that, when combined using a Boolean logic function, shows strong association with the gene expression levels.

2 Results

We have performed Mutational Genomics of a set of 97 retrovirally induced splenic lymphomas in $p19^{ARF-/-}$ (n=31), $p53^{-/-}$ (n=19) and wt (n=53) mice. The retroviral insertion sites found in these tumors have been published previously[1] [18]. Gene expression data were obtained using the Illumina MouseWG6-V2 beadchips. A detailed

[1] Available at `http://mutapedia.nki.nl`

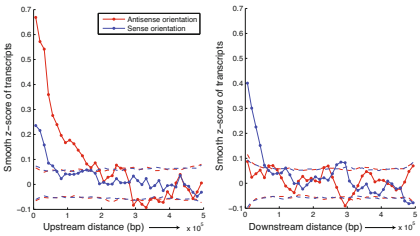


Fig. 2. Insertion alignment plots showing effect of insertions on transcription. The solid lines represent the smoothed z-scores of transcripts with insertions upstream (left) or downstream (right). Distance is relative to transcription start sites. Insertions were also split according to their orientation relative to the transcripts with red lines indicating 'anti-sense' insertion effects (insertion orientation opposite to transcript orientation). The inverse holds for the lines. The dashed lines reflect the 5% significance threshold, obtained by permutation.

description of the preprocessing of the data can be found in the Methods section and the Supplementary material.

2.1 Insertions Affect Local Transcription

We first investigated the local effect of the insertional mutations on transcription. Figure 2 shows a genome-wide alignment of all insertions in the dataset. A point in this figure at (d, z) represents the average z-score (z) of the expression of all genes in a bin d basepairs removed from the insertion. Panel A and B show the result for genes with upstream and downstream insertions, respectively, with different colors indicating insertion orientation relative to the transcript.

Figure 2 reveals that, on a global level, a clear effect of the insertions on the local transcription is present but that this effect is dependent on distance. Furthermore, it can be seen that antisense insertions result in a higher average expression, indicating a strong effect on local transcription, when their relative position to the transcript is upstream. Conversely, sense insertions seem to have a stronger effect in case they are positioned downstream of the transcript. These observations are consistent with previously described mechanisms through which retroviruses act on their targets [5, 7, 16]. For this reason we decided to implement a set of literature derived rules that map insertions to their putative target transcripts based on their relative position and orientation (see Supplements for details). This provides a mapping of all insertions in a given genomic locus to a unique identifier.

2.2 Mutational Genomics Reveals eQMLs

Association Inference. After normalization and selection of the most highly variable probes, probes were hierarchically clustered using a stringent correlation distance cutoff. This yielded 6228 clusters, henceforth referred to as gene clusters. For gene clusters containing multiple genes (1177 cases) cluster centroids were determined by taking the mean across the expression profiles.

To determine the Boolean insertion matrix (representing the insertion loci, see Figure 1D), all insertions were mapped to their target transcripts according to the literature derived rules. Each transcript represents one column of the Boolean insertion matrix and is determined by recording TRUE in case a tumor contains a mapped insertion or

FALSE in case it does not. Only columns with at least three mapped insertions were retained. This resulted in a Boolean matrix with 200 unique columns representing the insertion loci. To incorporate possible interactions with the genotype status of these tumors ($p19^{\mathrm{ARF}-/-}$, $p53^{-/-}$ or wt), we included three additional columns representing the three genotypes.

To measure association between the insertion loci (or combination of insertion loci) and the gene clusters we used a standard t-score. For each of the gene clusters we determined the single best locus with the strongest positive and negative association, as well as the best possible combination of loci for each of the 24 Boolean network topologies (Figure S1). Solutions with a permutation based p-value smaller than 0.001 were retained. In case multiple solutions for a single gene cluster remained, a rank aggregation approach (described in detail in the supplement), combining several measures of significance and biological relevance, was used to choose the most relevant model.

Interaction Network. Using this approach, we find significant ($p < 0.001$) single locus and mutilocus associations for 137 gene clusters (174 genes). A Cytoscape plot of these interactions is given in Figure S5. For 88 of the gene clusters, a single locus model, i.e. inserts at a single locus, was sufficient to obtain a significant segregation of the expression measurements. On the other hand, for 49 cases a more complex association was required to obtain a significant association (20 2-input networks and 29 3-input networks). Interestingly, the type of logic that was used in this set of significant interactions was depleted of AND logic. In fact, it was observed that AND logic generally showed poor association (irrespective of the p-value), suggesting that co-occurrence of insertions (i.e. insertions co-occurring in the same tumor, captured by AND logic) is not a common mechanism in regulating transcriptional activity.

cis-eQTMLs. Strong cis-associations, for which an insertion locus is associated with a proximal target transcript, are observed for insertions mapped to *Rras2*, *Ccnd1*, *Gfi1* and *Notch1*. In many other cases, direct association on the transcriptional level between insertions and their predicted targets is more subtle, i.e. the expression changes are very small, and fail to exceed the array noise. In other cases insertions may affect translation instead of transcription, and hence may not be detected in this analysis.

It is possible that the use of alternative routes of deregulating nearby genes dilutes the observed cis-association. This means that the absence of a mutation is no longer necessarily associated with low expression. A clear example of such a case is the expression of the *Myc* oncogene, which was found to be expressed (log2 expression level > 7) in 88 of the 97 tumors, while it harbored an insertion only in 51 tumors (Figure 3). This suggests that, in cases where an insertion near *Myc* is lacking, *Myc* is upregulated by other mechanisms. For most of the tumors in which *Myc* remains unexpressed, insertions near *Mycn* are observed. Indeed, our results reveal a strong negative association between insertions near *Mycn* and *Myc* expression (Figure 3). A plausible explanation for this observation is that *Mycn* insertions are functionally equivalent to insertions in the *Myc* locus, a mechanism which has been identified in human leukemias and lymphomas as well [19].

Genotype Interactions. By including three Boolean profiles representing the genotype we are able to retrieve genotype specific expression changes, as well as expression

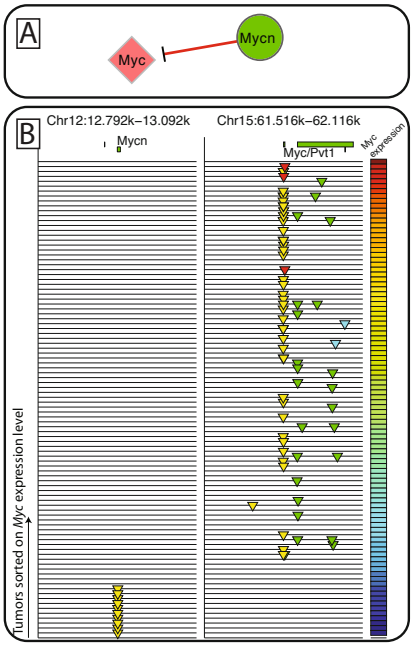


Fig. 3. Association between *Mycn* insertions and *Myc* expression. A) The red diamond-shaped node represents the gene cluster containing, in this case, a single probe for *Myc*. The green circular node represents the insertion locus for *Mycn*. **B)** Locus plot of insertions in the *Mycn* locus and the *Myc* locus. Green (yellow) triangles denote positively (negatively) oriented oriented insertions that according to the literature rules were mapped to *Mycn/Myc*. Red (cyan) triangles denote positively (negatively) oriented insertions that were not mapped to a target gene. The color bar on the right represents expression levels of the *Myc* probe. Tumors were sorted based on the expression level of *Myc*.

changes that are due to putative interaction between genotype and one or two insertion loci (Figure 4). In addition to the probe for *p53* itself, many other well characterized targets of *p53* and *p19* were found among the direct associations identified by our analysis. More specifically, increase of *Cdkn2a* (*p19*^{ARF} isoform) expression is associated with the *p53*$^{-/-}$ tumors, suggesting a feedback loop mechanism compensating for the loss of *p53*. Interestingly, low expression of the *p16INK4a* isoform is found to be associated with wild-type tumors only, suggesting loss of the *p19/p53* pathway permits lymphoma development in the presence of increased p16 expression. Other known direct interactions include: *Bax* [20], *Cdkn1a* (p21) [21] and *Ccng1* (CyclinG1) [22]) all of which are induced by p53. These examples demonstrate the robustness of our methodology.

A more complex association between genotype and transcript level was found in the case of *Usp18*, a gene which has been implicated in human non-small-cell lung cancer [23]. A 3-input network with the wild-type status, *p19*$^{\mathrm{ARF}-/-}$ status and the *Nfkb2/Sufu* locus was found to be negatively associated with low *Usp18* transcript levels. This network can be simplified to a 2-input OR network with *p53*$^{-/-}$ status and the *Nfkb2/Sufu* locus as inputs and a positive association with *Usp18* expression (Figure S4). Indeed, the *p53*$^{-/-}$ status was found to be strongly associated with elevated *Usp18* levels. However, in a substantial number of wild-type and *p19*$^{\mathrm{ARF}-/-}$ tumors elevated expression was also observed. Interestingly, the CAL network offers a partial explanation for this, since it reveals that three of the non-*p53*$^{-/-}$ tumors with high *Usp18* expression harbored insertions in the *Nfkb2/Sufu* locus. From this observation the interesting hypothesis can be derived that insertions near *Nfkb2/Sufu* offer an alternative to the loss of *p53* in upregulating *Usp18*.

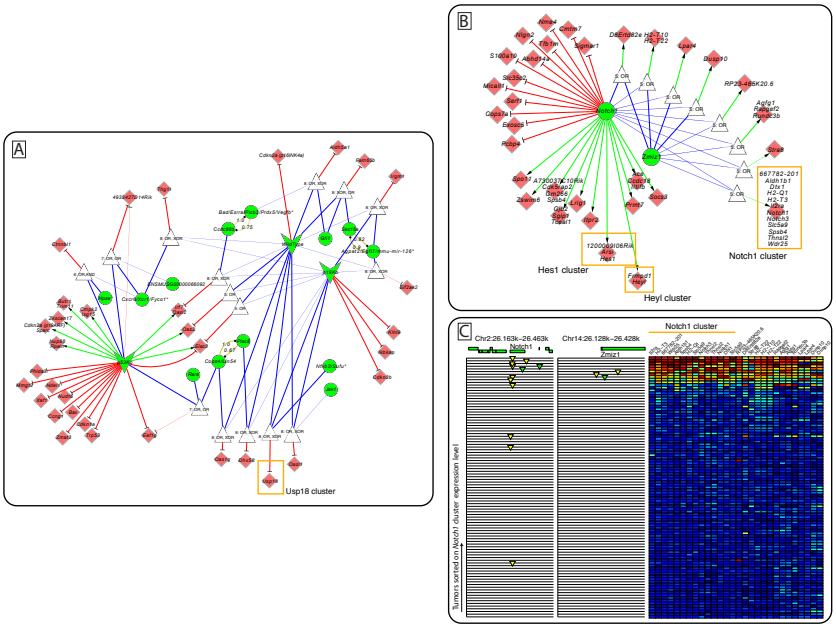


Fig. 4. Cytoscape interaction diagrams of the interactions with the genotype (A) and *Notch1* B) status. Green V-shaped nodes, green circular nodes, red diamond-shaped nodes represent the genotype status, insertion loci and gene clusters, respectively. The white triangles denote CAL networks, with the logic functions used specified in text. The number in the network nodes refer to the supplementary table. Green and red links represent positive and negative associations, respectively. The yellow links indicate proximal insertion loci, that share some of the mapped insertions. The numbers on these links indicate the fraction of insertions that are shared. In case the nodes are labeled with a (*), some genes were omitted from the complete list of putative targets for readability. Putative targets were only omitted in case literature revealed poor evidence for involvement in cancer or cell-functions like apoptosis or cell-cycle. A full list of putative targets is available in the online material (see Supplements for details). **C)** Locus plot of the *Notch1* and *Zmiz1* loci. For an explanation of the symbols see Figure 3. Only the probes at the output of a 2-input OR network with *Notch1* and *Zmiz1* are shown. Expression values were *z*-normalized to allow for comparison between probes.

Regulatory Hubs. The discovered interactions reveal that *Gfi1* and *Notch1* are clear hubs, and insertions in their vicinity are associated with expression of many transcripts. Interestingly, both genes have well established roles in cancer and moreover are known transcriptional regulators.

Gfi1 encodes a nuclear zinc finger protein and is recognized to have different complex and cell context specific roles. In lymphoid cells, however, GFI1 is a known transcriptional repressor. This is consistent with the predominantly inhibitory interactions revealed by our analysis. The literature provides evidence for some of the putative regulatory interactions. An interesting example is negative association between inserts near *Gfi1* and transcript levels of *Btg1*. Human BTG1 is a known tumor suppressor and member of an anti-proliferative gene family that regulates cell growth and

differentiation [24]. It has been implicated in acute lymphoblastic leukemia (ALL) [25] and non-hodgins lymphoma [26]. Association between *Gfi1* and *Btg1* activity may be explained as it was found that BTG1 is regulated by CEBPA [27], which, in turn, is a known target of GFI1 in T lymphocyte (Jurkat) cells [28].

Figure 4A shows the interaction diagram of associations of the *Notch1* locus and gene expression of multiple genes. In addition, the associations of a 2-input OR of *Notch1* and *Zmiz1* are shown. *Notch1* is a member of the family of NOTCH receptors, that operate both as recipients of extracellular signals at the cell surface and as transcription factors regulating gene expression in the nucleus. In its role as transcription factor, NOTCH1 forms a transcriptional activator complex and activates genes of the enhancer of split locus. Notably, *Hes1*, hairy and enhancer of split 1, and *Heyl*, a member of the hairy and enhancer of split-related (HESR) family, are both among the associated transcripts identified by our analysis. Both proteins have been implicated in cancer, and specifically implicated as targets of NOTCH signalling [29].

Using Chip-chip data previously published [30] of NOTCH1 and HES1 DNA binding in human T cell ALL cells [30], we checked if the orthologs of the *Notch1* target transcripts identified in our study were among the list of NOTCH1 bound genes. We found that 5 of the 23 *Notch1* targets with human orthologs were among the NOTCH bound target list (COPS7A, EXOSC5, HES1, ITPR2 and TFB1M). Since *Hes1* was among our *Notch1* targets, and it is possible that *Notch1* acts upon its targets through *Hes1*, binding of HES1 may explain the associations observed with *Notch1* mutations [31]. Therefore, we also checked for overlap of human orthologs of *Notch1* targets and the Chip-chip results of HES1 binding. In this way suggestive evidence for three additional interactions was found (CDK5RAP2, PRMT7 and TCEAL1).

Multi-locus eQMLs Reveal Alternative Pathways. Although *Notch1* insertions are found almost exclusively in tumors with elevated transcripts levels of *Notch1*, three tumors remain without *Notch1* insertions (Figure 4). One CAL network combines the *Notch1* locus with insertions in the *Zmiz1* locus. Insertions in the *Zmiz1* locus occur in tumors with elevated *Notch1* levels and two of these occur in tumors without insertions in the *Notch1* locus. Moreover, *Zmiz1* insertions are exclusively observed for tumors with elevated *Notch1*. A hypothesis worth exploring further is therefore that *Zmiz1* operates upstream of *Notch1* and, in case of the absence of a *Notch1* mutation, is able to upregulate *Notch1*.

3 Discussion

We propose mutational genomics, an approach to delineate transcriptional regulatory interaction networks in cancer by searching for associations in mutation data and gene expression measurements obtained from the same sample. When performed for a set of 97 lymphoid splenic tumors, an interaction network comprising 60 insertionally targeted loci and 174 putative target transcripts results. Because selective pressure exerted by the tumor growth enriches for loci with causal implications for tumorigenesis, many interactions in cancer related pathways were discovered.

A number of well characterized interactions were found, such as the association between loss of *p53* and reduced *Bax*, *Cdkn1a* and *Ccng1* levels. Known transcriptional regulators *Gfi1* and *Notch1*, both of which have established roles in tumorigenesis, were

found to be associated to differential expression of many transcripts, suggesting a master regulator role for these genes in lymphomagenesis. The targets of insertions near *Notch1* included many genes whose promotors were found to be bound by NOTCH1 and or HES1 in human T cell ALL.

In addition to single locus associations, more complex associations were identified by inferring CAL networks, i.e. Boolean combinations of insertion loci. This revealed a possible role for insertions in the *Nfkb2/Sufu* locus in upregulating *Usp18* expression. Similarly, it was found that two of the tumors that did not appear to bear an activating *Notch1* mutation, harbored insertions in the *Zmiz1* locus, possibly explaining the elevated *Notch1* expression in these tumors. From this the hypothesis can be formulated that *Zmiz1* functions upstream of *Notch1*. This illustrates the potential of mutational genomics as a powerful way of generating hypotheses that can be validated in the lab.

While in this study we focused on retroviral insertional mutagenesis, transposon based insertional mutagenesis may be similarly suitable for mutational genomics [32]. This would greatly increase the number of tissues and tumor types in which mutational genomics can be employed, and thus increase the scope of this approach.

4 Materials and Methods

Animal Experiments. All animal experiments were done conform to national regulatory standards and are approved by the Animal Experiments Committee (DEC) of the Netherlands Cancer Institute (approval ID: OZP 02029).

Gene Expression Preprocessing. Gene expression measurements were obtained using the Illumina MouseWG6-V2 beadchips, and were normalized using VST and RSN. Probes without a map position were discarded. Only highly variant probes (within the top 25 percentile) were retained. Hierarchical clustering (complete linkage, correlation distance, distance threshold of 0.2) was employed to combined strongly correlated genes, resulting in 6261 clusters. A clipping filter was applied as described [17], to limit the effect of strong outliers, affecting 625 gene clusters. Finally, gene clusters for which the best possible split in two groups based on the t-score resulted in highly unbalanced class distribution (smallest class size of 3 or smaller), were removed. Altogether, this resulted in 6228 gene clusters that were used in the association analysis.

Determining Insertion Loci. The effect of insertions on the nearby targets is dependent on the relative position and orientation of the target transcript as well as the orientation of the viral integration [5, 7, 16]. To exploit this information, we have employed a rule-based mapping (RBM) procedure [33]. RBM associates each insertion to one or more putative target transcripts based on a set of rules that were distilled from literature (a more comprehensive description of RBM is given in the Supplements). The unique list of transcripts that follows from this procedure is used to generate binary profiles that, for each tumor, indicate if a transcript is a putative target. We observed that for proximal transcripts frequently the same binary profile results. These were therefore combined into a single profile. Insertion target profiles that contained transcript-insertion associations in more than three tumors were considered in the analysis and served as inputs for the association inference.

CAL Network Inference. CAL network inference has been described in detail [17]. Briefly, given some Boolean network topology $\mathcal{B}$, the objective is to find the combination of loci such that the association between the network output and some gene expression vector is optimal. Equivalently, we solve the following:

$$\underset{\mathbf{L}}{\operatorname{argmax}} f(\mathcal{B}(\mathbf{L}), \mathbf{g}), \tag{1}$$

where $\mathbf{L}$ is a $T \times N$ Boolean matrix containing N input loci of length T, $\mathcal{B}$ is a Boolean function that maps the N input loci to a single Boolean vector, $\mathbf{g}$ is a vector containing the expression values for some gene and f is an association measure. Here, we use the t-statistic as the association measure. In the tail of the t-distribution an approximation of the t-score exists that can be optimized, using a branch-and-bound algorithm, in a fraction of the time required to optimize the real t-score.

To apply the CAL network inference approach to a dataset with $\sim$100 samples, some modifications to the original implementation of this method [17] were made in order to improve scalability further. All modifications are described in the Supplementary material.

CAL Network Significance. We solve Equation 1 for each gene cluster and for a range of 24 network topologies. The topologies are given in Figure S1. For each gene cluster-network topology combination a p-value can be obtained. The following procedure is performed to obtain the necessary null-distributions for each network topology separately. All 6228 gene clusters are permuted 90 times by shuffling the order of the clusters' gene expression values. This results in a total of 560k random permutations. For each permutation the CAL network search is performed, using the same parameter settings as were used on the real data. This results in 560k t-scores. The CAL network algorithm only produces reliable solutions above a certain tolerance level, which for these data was set to $t = 7.5$. We therefore calculate a piecewise cumulative distribution function (CDF). Below the tolerance level the CDF is set to zero, since in this region t-scores are not accurate. Above the tolerance, we use the empirical estimate of the CDF. A pseudocount is included to prevent p-values of zero.

In many cases it is possible to find strong (and significant) associations between the mutation data and gene expression using several network topologies. In order to select the most biologically relevant model, we rank all solutions based on several other measures of significance and biological relevance. These measures include: 1) the p-value improvement compared to the lowest p-value obtained for networks with fewer inputs, 2) the number of inputs of the network topology, 3) the coverage of the truth table of the network topology, 4) the number of samples in the smallest class. Average Borda ranking is used to aggregate ranks from these four measures [34]. Only solutions that receive the highest rank are reported.

Acknowledgements. This work was part of the BioRange programme of the Netherlands Bioinformatics Centre (NBIC), which is supported by a BSIK grant through the Netherlands Genomics Initiative (NGI). This research has partially been supported by the Dutch Life Science Grid initiative of NBIC and the Dutch e-Science Grid BiG Grid, SARA - High Performance Computing and Visualisation.

References

1. Hanahan, D., Weinberg, R.A.: Hallmarks of cancer: the next generation. Cell 144, 646–674 (2011)
2. van't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A.M., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature 415, 530–536 (2002)
3. van de Vijver, M.J., He, Y.D., van't Veer, L.J., Dai, H., Hart, A.A.M., et al.: A gene-expression signature as a predictor of survival in breast cancer. N Engl. J. Med. 347, 1999–2009 (2002)
4. Sørlie, T., Perou, C.M., Tibshirani, R., Aas, T., Geisler, S., et al.: Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. Proc. Natl. Acad. Sci. USA 98, 10869–10874 (2001)
5. Kool, J., Berns, A.: High-throughput insertional mutagenesis screens in mice to identify oncogenic networks. Nature Reviews Cancer 9, 389–399 (2009)
6. Kool, J., Uren, A.G., Martins, C.P., Sie, D., de Ridder, J., et al.: Insertional mutagenesis in mice deficient for p15ink4b, p16ink4a, p21cip1, and p27kip1 reveals cancer gene interactions and correlations with tumor phenotypes. Cancer Res. 70, 520–531 (2010)
7. Uren, A.G., et al.: Retroviral insertional mutagenesis: past, present and future. Oncogene 24, 7656–7672 (2005)
8. Mikkers, H., Berns, A.: Retroviral insertional mutagenesis: tagging cancer pathways. Adv. Cancer Res. 88, 53–99 (2003)
9. Jansen, R.C., Nap, J.P.: Genetical genomics: the added value from segregation. Trends Genet. 17, 388–391 (2001)
10. Gerrits, A., Dykstra, B., Otten, M., Bystrykh, L., de Haan, G.: Combining transcriptional profiling and genetic linkage analysis to uncover gene networks operating in hematopoietic stem cells and their progeny. Immunogenetics 60, 411–422 (2008)
11. Li, J., Burmeister, M.: Genetical genomics: combining genetics with gene expression analysis. Hum. Mol. Genet. 14(spec. 2), R163–R169 (2005)
12. Schadt, E.E., Monks, S.A., Drake, T.A., Lusis, A.J., Che, N., et al.: Genetics of gene expression surveyed in maize, mouse and man. Nature 422, 297–302 (2003)
13. Bystrykh, L., Weersing, E., Dontje, B., Sutton, S., Pletcher, M.T., et al.: Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. Nat. Genet. 37, 225–232 (2005)
14. Gerrits, A., Li, Y., Tesson, B.M., Bystrykh, L.V., Weersing, E., et al.: Expression quantitative trait loci are highly sensitive to cellular differentiation state. PLoS Genet 5, e1000692 (2009)
15. Erkeland, S.J., Verhaak, R.G.W., Valk, P.J.M., Delwel, R., Löwenberg, B., et al.: Significance of murine retroviral mutagenesis for identification of disease genes in human acute myeloid leukemia. Cancer Res. 66, 622–626 (2006)
16. Jonkers, J., Berns, A.: Retroviral insertional mutagenesis as a strategy to identify cancer genes. Biochim. Biophys. Acta 1287, 29–57 (1996)
17. de Ridder, J., Gerrits, A., Bot, J., de Haan, G., Reinders, M., et al.: Inferring combinatorial association logic networks in multimodal genome-wide screens. Bioinformatics 26, i149–157 (2010)
18. Uren, A.G., Kool, J., Matentzoglu, K., de Ridder, J., Mattison, J., et al.: Large-scale mutagenesis in p19(arf)- and p53-deficient mice identifies cancer genes and their collaborative networks. Cell 133, 727–741 (2008)
19. Hirvonen, H., Hukkanen, V., Salmi, T.T., Pelliniemi, T.T., Alitalo, R.: L-myc and n-myc in hematopoietic malignancies. Leuk Lymphoma 11, 197–205 (1993)
20. Chipuk, J.E., Kuwana, T., Bouchier-Hayes, L., Droin, N.M., Newmeyer, D.D., et al.: Direct activation of bax by p53 mediates mitochondrial membrane permeabilization and apoptosis. Science 303, 1010–1014 (2004)

21. Dulić, V., Kaufmann, W.K., Wilson, S.J., Tlsty, T.D., Lees, E., et al.: p53-dependent inhibition of cyclin-dependent kinase activities in human fibroblasts during radiation-induced g1 arrest. Cell 76, 1013–1023 (1994)
22. Komarova, E.A., Diatchenko, L., Rokhlin, O.W., Hill, J.E., Wang, Z.J., et al.: Stress-induced secretion of growth inhibitors: a novel tumor suppressor function of p53. Oncogene 17, 1089–1096 (1998)
23. Lam, D.C.L., Girard, L., Ramirez, R., Chau, W.S., Suen, W.S., et al.: Expression of nicotinic acetylcholine receptor subunit genes in non-small-cell lung cancer reveals differences between smokers and nonsmokers. Cancer Res 67, 4638–4647 (2007)
24. Rouault, J.P., Rimokh, R., Tessa, C., Paranhos, G., Ffrench, M., et al.: Btg1, a member of a new family of antiproliferative genes. EMBO J. 11, 1663–1670 (1992)
25. van Galen, J.C., Kuiper, R.P., van Emst, L., Levers, M., Tijchon, E., et al.: Btg1 regulates glucocorticoid receptor autoinduction in acute lymphoblastic leukemia. Blood 115, 4810–4819 (2010)
26. Morin, R.D., Mendez-Lago, M., Mungall, A.J., Goya, R., Mungall, K.L., et al.: Frequent mutation of histone-modifying genes in non-hodgkin lymphoma. Nature 476, 298–303 (2011)
27. Tavor, S., Park, D.J., Gery, S., Vuong, P.T., Gombart, A.F., et al.: Restoration of c/ebpalpha expression in a bcr-abl+ cell line induces terminal granulocytic differentiation. J. Biol. Chem. 278, 52651–52659 (2003)
28. Duan, Z., Horwitz, M.: Targets of the transcriptional repressor oncoprotein gfi-1. Proc. Natl. Acad. Sci. U S A 100, 5932–5937 (2003)
29. Katoh, M., Katoh, M.: Integrative genomic analyses on hes/hey family: Notch-independent hes1, hes3 transcription in undifferentiated es cells, and notch-dependent hes1, hes5, hey1, hey2, heyl transcription in fetal tissues, adult tissues, or cancer. Int. J. Oncol. 31, 461–466 (2007)
30. Margolin, A.A., Palomero, T., Sumazin, P., Califano, A., Ferrando, A.A., et al.: Chip-on-chip significance analysis reveals large-scale binding and regulation by human transcription factor oncogenes. Proc. Natl. Acad. Sci. U S A 106, 244–249 (2009)
31. Dudley, D.D., Wang, H.C., Sun, X.H.: Hes1 potentiates t cell lymphomagenesis by up-regulating a subset of notch target genes. PLoS One 4, e6678 (2009)
32. Mattison, J., van der Weyden, L., Hubbard, T., Adams, D.J.: Cancer gene discovery in mouse and man. Biochim. Biophys. Acta 1796, 140–161 (2009)
33. de Jong, J., de Ridder, J., van der Weyden, L., Sun, N., van Uitert, M., et al.: Computational identification of insertional mutagenesis targets for cancer gene discovery. Nucleic Acids Res 39, e105 (2011)
34. Lin, S.: Rank aggregation methods. Wiley Interdisciplinary Reviews: Computational Statistics (2010)

Outlier Gene Set Analysis Combined with Top Scoring Pair Provides Robust Biomarkers of Pathway Activity

Michael F. Ochs[1,⋆], Jason E. Farrar[2], Michael Considine[1], Yingying Wei[3], Soheil Meschinchi[4], and Robert J. Arceci[5]

[1] The Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University, Baltimore, MD, USA
mfo@jhu.edu
[2] College of Medicine, University of Arkansas for Medical Sciences, Little Rock, AR, USA
[3] The Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD, USA
[4] Fred Hutchinson Cancer Research Center, Seattle, WA, USA
[5] Ronald A. Matricaria Institute of Molecular Medicine, Phoenix Children's Hospital, Phoenix, AZ, USA

Abstract. Cancer is a disease driven by pathway activity, while useful biomarkers to predict outcome (prognostic markers) or determine treatment (treatment markers) rely on individual genes, proteins, or metabolites. We provide a novel approach that isolates pathways of interest by integrating outlier analysis and gene set analysis and couple it to the top-scoring pair algorithm to identify robust biomarkers. We demonstrate this methodology on pediatric acute myeloid leukemia (AML) data. We develop a biomarker in primary AML tumors, demonstrate robustness with an independent primary tumor data set, and show that the identified biomarkers also function well in relapsed AML tumors.

1 Introduction

The development of cancer is known to be driven by deregulation of several biological processes, referred to as the Hallmarks of Cancer [4], and loss of control of each process is required for the development of lethal cancers in almost all cases. Regulation of most of these Hallmarks relies on proper functioning of cell signaling pathways [5], which comprise sets of signaling proteins, primarily kinases and phosphatases, that work to transduce a signal through a cell by means of post-translational modifications of proteins. The deregulation of any single pathway can be driven by a mutation or other change in a single protein within the pathway [11].

⋆ Corresponding author.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 47–58, 2013.

1.1 Outlier Gene Set Analysis

The dominance of pathways over genes in the etiology of cancer creates a problem for statistical analysis that focuses on determining global behaviors in cancers in general or types of cancer in particular. Since loss of regulation of a pathway is the critical event, but global measurements focus on genes and the proteins they encode, there is a mismatch in the statistic (based on data from genes) and the effect (based on pathway deregulation). This suggests a need for a pathway-based statistic for use in cancer studies.

The first issue to resolve is that any given gene in a subtype of cancer is likely to be affected in only a small fraction of individuals, since there are many potential genes that may drive pathway deregulation. For example, the well-studied RAS-RAF pathway may become deregulated through overexpression of the EGFR receptor, mutation of the RAS, RAF, or MAPK genes, or mutation or overexpression of the MYC transcriptional regulator. Any individual is likely to have only one such change, and no single change is likely to rise above $\sim 50\%$ of cases, with most lying between 5% and 15%. This limits the value of standard statistical tests, such as t-tests or ANOVA analyses.

However, outlier analysis, such as Cancer Outlier Profile Analysis [9], provides a method to identify those genes that are deregulated in only a subset of individuals. While useful, this alone will not provide the required identification of deregulated pathways, although it should provide an indication of significance of the individual pathway members. With Gene Set Analysis (GSA) we can integrate these estimates of significance to provide an overall estimate of pathway significance on a global scale, which we refer to as Outlier Gene Set Analysis (OGSA). This provides a global estimate of pathway deregulation in cancer subtypes.

1.2 Pathway-Based Top Scoring Pairs

The fundamental measurements we make clinically remain linked to genes, not pathways. This complicates the development of diagnostic tests for the drivers in cancer, the pathways. In general, we visualize the deregulation of the pathway through heatmaps and other data-driven visualization tools. However, these provide poor clinical utility as the results change with addition of data, making them inappropriate for clinical tests that must deduce a probability from an isolated measurement, and they have been shown to be strongly platform dependent, increasing the potential cost and reducing the opportunity for innovation.

In order to create a method that could identify robust potential biomarkers, the multigene signature generated from discriminant analysis can be replaced by pairs of genes that change their relative level of expression [14], known as a Top Scoring Pair (TSP) [1]. In TSP, the statistic of interest is how well the measurements on a pair of genes distinguish two classes, relying on the inversion of the values of measurements between classes. This provides a normalization-independent approach that makes switching measurement technologies far more likely to succeed [12]. However, a limitation of TSP is that it searches through all

possible TSPs, introducing the potential of chance identification of pairs that are not robust and fail to validate. Instead, we build a pathway TSP set by limiting the domain for generating TSPs to pathways of interest in the set of statistically significant pathways generated by OGSA. In this way, we focus the methodology on biologically-motivated gene sets, more suitable for clinical development than unbiased discovery.

1.3 Pediatric AML and the TARGET Initiative

Acute Myeloid Leukemia (AML) is a cancer of the blood affecting roughly $15,000$ individuals per year in the USA, and childhood patients show $\sim 60\%$ five year survival. However, the outcomes are highly dependent on karyotype-defined subtype, and initiatives to improve care for pediatric patients have led to broad molecular studies through the NCI Therapeutically Applicable Research to Generate Effective Treatments (TARGET) initiative.

1.4 Outline of Paper

In this paper, we describe the methodology in sections 2.1 and 2.2 together with the analysis of the AML data in section 2.3. In section 3, we show that OGSA of TARGET promoter methylation data identified the Hedgehog signaling pathway and the Cytochrome P450 metabolic pathway as highly epigenetically deregulated in pediatric AML. Using only genes associated with these pathways for the development of a set of TSPs, we demonstrate that we obtained a robust signature of pathway deregulation that was significant in an independent data set and also significant in samples from individuals whose cancer relapsed. Importantly, this suggests a novel therapeutic strategy in these patients and provides a potential treatment biomarker for this therapy.

2 Methods

Overall we adopted a number of key methodologies developed for identifying outlier genes and generating robust TSPs. We integrated these methods into a pathway-centric statistical approach that leverages outlier statistics to generate pathway statistics through OGSA and generates TSPs related to key pathways.

2.1 Outlier Gene Set Analysis

The standard method employed in cancer research for outlier analysis is Cancer Outlier Profile Analysis [9], which generates statistics by comparing the outlier distributions to an empirical null generated by permutation of class labels. However, this is computationally expensive and, importantly, we required only the rank of the genes and not their significance, since we utilized a rank-based gene set test (see below). Thus, we generated statistics using a modification that permits rapid p-value estimation, although this estimation is in general less reliable than that generated by a permutation test.

Each observation in a case sample was compared to the empirical distribution of expression values of the same gene for control samples following a ranksum methodology [3]. For gene g, we calculated the right-tail empricial p-value as

$$\hat{p}_{gt} = \frac{1}{N_{p0}} \sum_{i=1}^{N_{p0}} I(X_{gt} \leq X_{gi}) \tag{1}$$

where we indexed the control samples with i and the case samples by t with N_{p0} control samples and N_{p1} case samples. The corresponding left-tail empirical p-value was calculated as

$$\hat{p}_{gt} = \frac{1}{N_{p0}} \sum_{i=1}^{N_{p0}} I(X_{gt} \geq X_{gi}). \tag{2}$$

For both cases, we generated a $G \times N_{p1}$ matrix of empirical $\hat{p}$-values for each gene as an outlier in each case sample.

We modified these equations slightly in this study to incorporate biological knowledge of the impact of changes in methylation. Because cancers often show global methylation changes involving loss of intergenic methylation and increased methylation near genes, including areas measured by array technologies, it is not unusual for almost all tumor samples to show a slight increase in methylation in gene promoters relative to normal samples. However, these small methylation changes are not meaningful biologically, as they are not enough to drive changes in expression of the genes. As such, we modified Equations 1 and 2 by replacing X_{gi} by $X_{gi} + 0.1$ and $X_{gi} - 0.1$ respectively. Effectively, we counted outliers only when there was at least a 10% change in the level of methylation.

To generate rank statistics, we converted the $\hat{p}$-values to an indicator of significance by testing them against a Bonferroni corrected $\alpha = 0.05$ by

$$\hat{m}_{gt} = I(\hat{p}_{gt} \leq \frac{\alpha}{N_{p1}}) \tag{3}$$

where 1 indicates significant at level α and 0 indicates insignificant. The rank statistic was the sum of the indicator across all case samples, effectively ranking genes from N_{p1} to 0.

We analyzed these rank statistics using a mean rank gene set enrichment test [10], as provided in the limma R package [13], comparing the statistics of the genes in a gene set to genes outside the set. The mean-ranks of the test statistics for the genes were used for comparison, which matched our use of only the ranks of genes from outlier analysis. Gene sets were defined by the KEGG and BioCarta pathways [6] and final p-value estimates on the pathways were corrected for multiple testing using the Bonferroni method.

2.2 Pathway-Based Top Scoring Pairs

The OGSA method provides pathways that are significantly different between cases and controls, but it does not provide a suitable methodology for the

development of a test for a new sample. In order to generate such a test, we applied OGSA to highlight pathways of interest. We refined significant pathways by inspection, focusing on suitability for drug targeting or removal of pathways either universally modified or already addressed in treatment. We then used only the genes associated with the refined pathway list in TSP (i.e., those genes that define the gene set for this pathway in KEGG).

Table 1. Example TSP

	$G_i < G_j$	$G_i > G_j$	
Case	N_{TP}	N_{FN}	N_{case}
Control	N_{FP}	N_{TN}	N_{control}
	N_{callCase}	$N_{\text{callControl}}$	N

The choice of a TSP reduces to maximization of prediction in a Fisher two-way table, such that Table 1 provides the best possible predictive value for the measured levels G, here promoter methylation, of two genes i and j, where the relative levels of these genes determines the result of the test, with $G_i < G_j$ predicting a case and the inverse a control. The TSP is determined by finding the pair of genes that maximizes

$$\Delta_{ij} = |\frac{N_{TP}}{N_{\text{case}}} - \frac{N_{FP}}{N_{\text{control}}}|, \tag{4}$$

where N_{TP} is the number of true positives, N_{FN} is the number of false negatives, N_{FP} is the number of false positives, and N_{TN} is the number of true negatives. The total number of measurements is N, divided into N_{case} cases and N_{control} controls. As TSP does not always provide ideal separation due to the inherent complexity of the underlying biology, the extension to kTSP, where multiple TSPs vote on case or control status, is natural [14].

Here we used kTSP, as implemented in the R ktspair package [2]. We generated five TSPs in our training set for voting on status of the samples.

2.3 Analysis of TARGET Methylation Data

We applied the OGSA and TSP methods to a set of samples from the NCI TARGET initiative. The data comprised 192 diagnostic samples of pediatric AML, 192 remission samples from the same patients after frontline treatment, and 46 relapse samples from those patients with a recurrence of AML. All measurements were made with Illumina HumanMethylation27 BeadChip arrays, and beta values (percent methylation) were generated from U and M probes. Methylation estimates showing low variance across all samples were removed, leaving 19999 promoter methylation estimates associated with 11871 genes. A training set of diagnostic and remission samples was generated from 96 patients by choosing

roughly 50% of the samples of each karyotype in the data set. This balanced set was chosen to avoid biasing the training set to any particular diagnostic subtype, as different karyotypes have different outcomes in AML. Samples from the remaining 96 patients formed the test set, and an additional set of remission and relapse samples was generated based on the 46 relapse samples.

The OGSA method was applied to the training set and significant pathways were determined. For genes with multiple associated methylation probes, the probe with the highest mean methylation was retained. From the significant pathways, one driven by hypermethylation and one by hypomethylation were chosen based on the usefulness for drug targeting and metabolism as well as on their lack of being associated globally with all cancers. The use of one hypermethylation and one hypomethylation driven pathway was to increase the potential range of top-scoring pairs.

Five TSPs were generated from the probes associated with the genes assigned to the two pathways using the ktspair package applied to the training data set. These pairs were then used to vote on each sample, and the cutoff that maximized the predictive power of the pairs was used. These same pairs and cutoff were then applied to the training data set and to the relapse-remission data.

The key targetable pathway was also visualized using a heatmap of the genes in the pathway. This permitted visual comparison of the separation of diagnostic samples from remission samples, as well as the separation of relapse and remission samples. To test whether the pathway associated with karyotype, separation of karyotype on the heatmap was also investigated; however, there was no correlation (heatmap not shown).

3 Results

We applied our methods to the TARGET AML data comprising 430 samples as discussed in the Methods section. We analyzed the three separate data sets, Training, Test, Relapse, as follows. We first performed outlier analysis on the Training data, ranking all genes based on their outliers according to the sum across all diagnosis samples (Equation 3). These gene ranks were used to generate a set of significant pathways from the KEGG and Biocarta pathways using OGSA. We focused on two pathways from this set, the KEGG Hedgehog Signaling and Cytochrome P450 Metabolism pathways, for reasons detailed below. Using genes from the Hedgehog pathway, we created heatmaps of the Training, Test, and Relapse data to visualize the separation of samples. Using only the Training data, we then created five TSPs from these pathways. We tested these TSPs on the Test and Relapse data, using an assumption that a vote for a diagnostic sample was equivalent to a vote for a relapse sample in the test.

3.1 Outlier Analysis and Gene Ranks

Outlier analysis according to Equation 3 provided outlier ranks for all genes. As shown in Figure 1, highly ranked genes showed substantial increases in

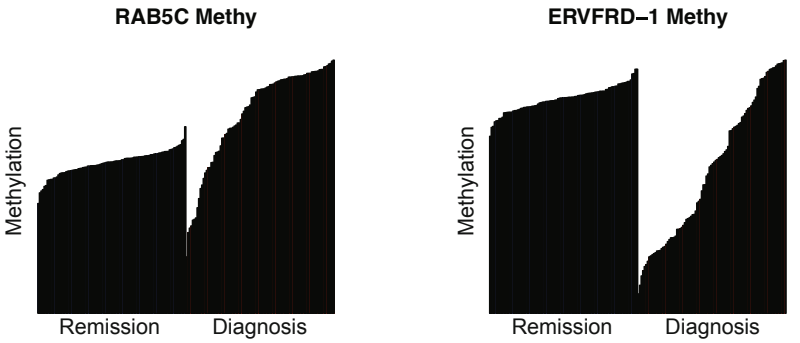


Fig. 1. The highest ranking right-tail and left-tail outlier genes from the Training data

methylation in diagnostic samples relative to remission samples. The top-ranked right-tail outlier gene, RAB5C, had 53 of 92 diagnostic samples called outliers, while the left-tail outlier gene, ERVFRD, had 48 of 92 diagnostic samples called outliers. The gene-based statistic is then provided by the rank from the gene with most outliers to the one with the fewest. The right-tail and left-tail rank lists were used in OGSA separately.

3.2 Significant Pathways from OGSA

The results of OGSA analysis of the KEGG and Biocarta pathway genes sets from the MSigDB database [8] are presented in Table 2. The p-values are Bonferroni corrected values from the mean rank gene set test. All pathways with significant p-values at the traditional $\alpha = 0.05$ are included in the table.

Many pathways in the right-tail analysis are seen in most GSA analyses of cancer data, including those involving focal adhesion and extracellular matrix receptor signaling (KEGG ECM Receptor Interaction, Cell Adhesion Molecules, Focal Adhesion pathways), pathways related to cancer (KEGG Neuroactive Ligand Receptor Interaction, Basal Cell Carcinoma, Pathways in Cancer pathways), and sets that appear significant in cancer studies due to the presence of genes related to integrin signaling and MAPK pathway activity (KEGG Dilated Cardiomyopathy and Arrhythmic Right Ventricular Cardiomyopathy pathways). These processes are deregulated in most cancers and do not provide novel insights to AML.

The pathways in the left-tail analysis are primarily involved in metabolism or immune responses. These pathways, in general, do not provide useful information for treatment and are generally hard to interpret in terms of cancer biology. Note that KEGG Neuroactive Ligand Receptor Interaction is significant in the left-tail analysis and the right-tail analysis, which indicates that methylation

Table 2. Significant KEGG and Biocarta Pathways

Right-Tail Outlier Results	p-Value
KEGG NEUROACTIVE LIGAND RECEPTOR INTERACTION	< 0.00001
KEGG ECM RECEPTOR INTERACTION	< 0.00001
KEGG HEDGEHOG SIGNALING PATHWAY	0.00001
KEGG BASAL CELL CARCINOMA	0.00032
KEGG PATHWAYS IN CANCER	0.00061
KEGG CELL ADHESION MOLECULES CAMS	0.00226
KEGG DILATED CARDIOMYOPATHY	0.00277
KEGG CALCIUM SIGNALING PATHWAY	0.01343
KEGG FOCAL ADHESION	0.01562
KEGG ARRHYTH RT VENTR CARDIOMYOPATHY ARVC	0.03704
Left-Tail Outlier Results	p-Value
KEGG COMPLEMENT AND COAGULATION CASCADES	< 0.00001
BIOCARTA COMP PATHWAY	< 0.00001
KEGG OLFACTORY TRANSDUCTION	< 0.00001
KEGG DRUG METABOLISM CYTOCHROME P450	0.00001
KEGG LINOLEIC ACID METABOLISM	0.00001
BIOCARTA CLASSIC PATHWAY	0.00004
KEGG METABOLISM OF XENOBIOTICS BY CYTOCHROME P450	0.00010
KEGG TYROSINE METABOLISM	0.00014
KEGG ARACHIDONIC ACID METABOLISM	0.00033
KEGG ETHER LIPID METABOLISM	0.00038
BIOCARTA LECTIN PATHWAY	0.00074
KEGG NEUROACTIVE LIGAND RECEPTOR INTERACTION	0.00301
KEGG STEROID HORMONE BIOSYNTHESIS	0.00348
KEGG RETINOL METABOLISM	0.01062
BIOCARTA INTRINSIC PATHWAY	0.03099

changes in the promoters of genes in this pathway include both hyper- and hypo-methylation.

The KEGG Hedgehog Signaling Pathway in the right-tail analysis attracted our attention, because Hedgehog signaling is known to be a driver of proliferation and antiapoptotic behavior, is involved in multiple cancers, is not typically associated with AML, and provides a potential target for treatment. To visualize the Hedgehog pathway methylation, we generated heatmaps of the samples, looking for separation of diagnostic, remission, and relapse samples (see Figure 2).

The Drug Metabolism Cytochrome P450 pathway and related Metabolism of Xenobiotics by Cytochrome P450 pathway in the left-tail analysis was suggestive given the importance of Cytochrome P450 in processing of therapeutic agents. The genes in this pathway coupled to the Hedgehog pathway genes provided a set of hypermethylated and hypomethylated genes suitable for creating a biomarker using TSP.

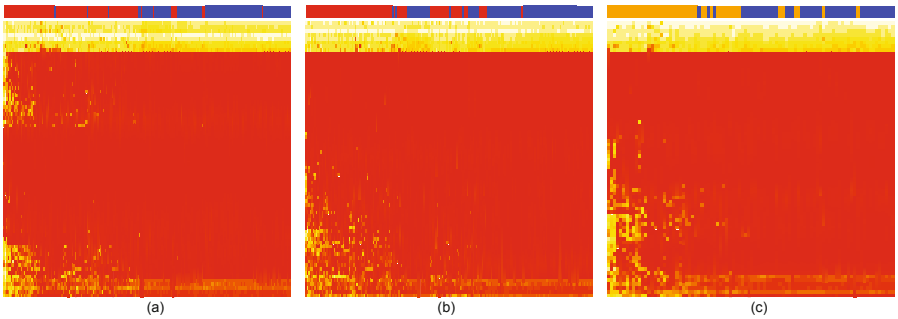


Fig. 2. Heatmaps of the methylation levels for promoters of genes in the KEGG Hedgehog pathway across patients in (a) the Training data, (b) the Test data, and (c) the Relapse data. In the top bar, blue indicates a remission sample, red a diagnostic sample, and orange a relapse sample. Genes are in rows and patients in columns. Yellow indicates high methylation ($\beta \to 1$) and red low methylation ($\beta \to 0$).

3.3 kTSP Classifiers for Hedgehog and Cytochrome P450 Pathways

In order to create a robust methylation signature for the Hedgehog and Cytochrome P450 pathways, we applied the kTSP algorithm to a subset of the Training data limited to promoter methylation levels of genes in the Hedgehog Signaling and Cytochrome P450 Metabolism pathways. We identified a set of 5 pairs that discriminate the diagnostic samples from the remission bone marrow samples (see Figure 3 where colors match the upper bar in Figure 2, so that blue is a remission sample and red a diagnostic sample). As seen in Table 3, this provided excellent prediction on the training set, with $p < 2.2 \times 10^{-16}$ and an odds ratio of 81 with a 95% confidence interval of $[28, 294]$.

Applying this signature to the Test data resulted in excellent prediction of diagnostic vs. remission samples, with $p < 2.2 \times 10^{-16}$, and an odds ratio of 128 with a 95% confidence interval of $[40, 563]$. Interestingly, the application of the same signature to the Relapse data set was also predictive, now of relapse vs. remission, with $p = 1.8 \times 10^{-6}$, and an odds ratio of 15 with a 95% confidence interval of $[4, 87]$. This suggests that relapse in pediatric AML may be partially driven by recurrence of methylation changes in the promoters of Hedgehog Signaling and Cytochrome P450 metabolism pathway genes, although the drop in sensitivity suggests that the relapse samples may be more diverse in this methylation than the diagnosis samples. Importantly, all tests show excellent Positive Predictive Values (94%, 95%, and 89% respectively), as is desirable for a test that could define treatment, since the vast majority of positive tests are related to positive pathway status.

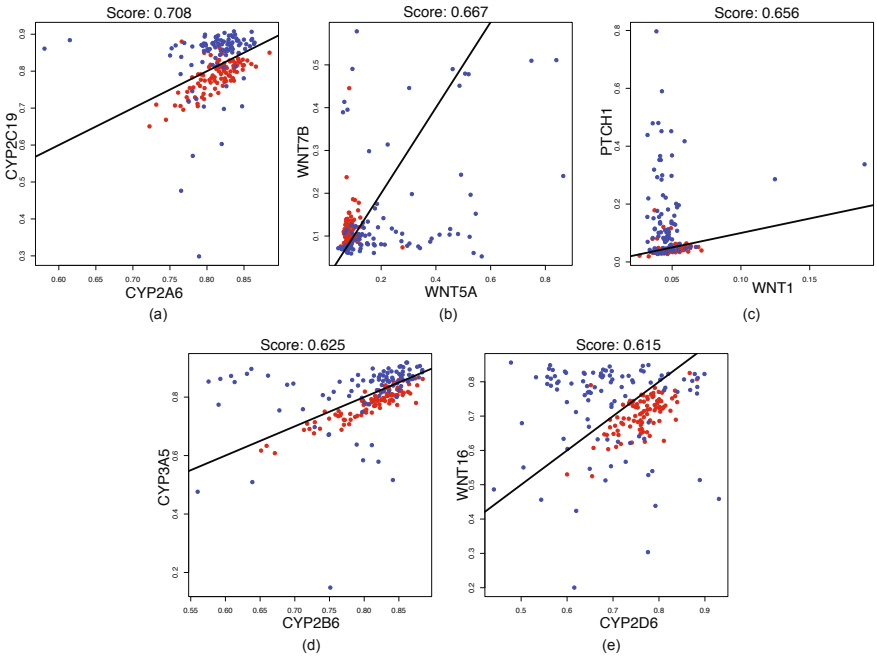


Fig. 3. The Five Top Scoring Pairs used to generate Table 3

Table 3. kTSP Classifier Performance

Training	Dx	Rm	**Test**	Dx	Rm	**Relapse**	Rl	Rm
Call Dx	79	5		82	4	Call Rl	24	3
Call Rm	17	91		14	92	Call Rm	22	43

4 Discussion

The coupling of outlier statistics, gene set analysis, and top scoring pair methods provides a solid methodology to identify deregulated pathways in cancer and to define a robust signature of their activity. We have shown that the method determines a robust marker, here comprising five TSPs, that validates in a completely novel data set, albeit one measured on the same platform at the same institution. Intriguingly, the marker does predict activity in the pathway in a subset of the relapse samples, suggesting both robustness of the marker and, potentially, that relapsed pediatric AML shows more heterogeneity than primary pediatric AML in Hedgehog activity. However, this suggestion is tempered by the low numbers and the known mismatch in karyotypes between primary and recurrent AML,

even though there was no correlation of Hedgehog pathway methylation with karyotype in primary tumors.

AML, specifically, and cancer in general, is difficult to treat effectively in most cases. Natural heterogeneity in response to treatment likely arises from both differences in molecular tumor characteristics and differences in systemic responses of individual patients [7]. Given this complexity, methods to define robust markers of potentially targetable pathways are extremely valuable to guiding treatment decisions, since the absence of cancer driver pathway activity should contraindicate targeted treatments for that pathway. The Positive Predictive Values (PPVs) from this test are therefore particularly promising, since a positive test is strongly indicative of pathway activity.

There remains a great need for more powerful, guided computational methods in cancer research and treatment. The complexity of the biological systems and a massive curse-of-dimensionality issue driven by small sample size coupled to genome-wide measurements of multiple molecular species present a formidable challenge requiring nonlinear modeling and novel computational learning techniques. It is likely the only viable approach will be to accept higher bias to reduce variance, and we have presented one such approach, where we limit our biomarker search based on statistically significant but knowledge-refined pathways.

Acknowledgements. MFO was funded by NIH/NLM R01LM011000 and NIH/NCI CCSG P30CA006973. MFO, JEF, MC, SM, and RJA were funded by the NIH/NCI U01 CA097452 National Childhood Cancer Foundation (TARGET). YW was partially funded by NIDCR RC1DE020324. RJA also received support from the endowed King Fahd Chair in Pediatric Oncology.

References

1. Geman, D., d'Avignon, C., Naiman, D.Q., Winslow, R.L.: Classifying gene expression profiles from pairwise mrna comparison. Statistical Applications in Genetics and Molecular Biology 3(1), 19 (2004)
2. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A.J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J.Y., Zhang, J.: Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5(10), R80 (2004)
3. Ghosh, D.: Discrete nonparametric algorithms for outlier detection with genomic data. Journal of Biopharmaceutical Statistics 20(2), 193–208 (2010)
4. Hanahan, D., Weinberg, R.A.: The hallmarks of cancer. Cell 100(1), 57–70 (2000)
5. Hanahan, D., Weinberg, R.A.: Hallmarks of cancer: the next generation. Cell 144(5), 646–674 (2011)
6. Kanehisa, M., Goto, S., Kawashima, S., Nakaya, A.: The KEGG databases at genomenet. Nucleic Acids Res. 30(1), 42–46 (2002)
7. Knox, S.S., Ochs, M.F.: Implications of systemic dysfunction for the etiology of malignancy. Gene. Regul. Syst. Bio. 7, 11–22 (2013)

8. Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdottir, H., Tamayo, P., Mesirov, J.P.: Molecular signatures database (MSigDB) 3.0. Bioinformatics 27(12), 1739–1740 (2011)
9. MacDonald, J.W., Ghosh, D.: COPA–cancer outlier profile analysis. Bioinformatics 22(23), 2950–2951 (2006)
10. Michaud, J., Simpson, K.M., Escher, R., Buchet-Poyau, K., Beissbarth, T., Carmichael, C., Ritchie, M.E., Schütz, F., Cannon, P., Liu, M., Shen, X., Ito, Y., Raskind, W.H., Horwitz, M.S., Osato, M., Turner, D.R., Speed, T.P., Kavallaris, M., Smyth, G.K., Scott, H.S.: Integrative analysis of RUNX1 downstream pathways and target genes. BMC Genomics 9, 363 (2008)
11. Parsons, D.W., Jones, S., Zhang, X., Lin, J.C., Leary, R.J., Angenendt, P., Mankoo, P., Carter, H., Siu, I.M., Gallia, G.L., Olivi, A., McLendon, R., Rasheed, B.A., Keir, S., Nikolskaya, T., Nikolsky, Y., Busam, D.A., Tekleab, H., Diaz Jr., L.A., Hartigan, J., Smith, D.R., Strausberg, R.L., Marie, S.K., Shinjo, S.M., Yan, H., Riggins, G.J., Bigner, D.D., Karchin, R., Papadopoulos, N., Parmigiani, G., Vogelstein, B., Velculescu, V.E., Kinzler, K.W.: An integrated genomic analysis of human glioblastoma multiforme. Science 321(5897), 1807–1812 (2008)
12. Price, N.D., Trent, J., El-Naggar, A.K., Cogdell, D., Taylor, E., Hunt, K.K., Pollock, R.E., Hood, L., Shmulevich, I., Zhang, W.: Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leiomyosarcomas. Proc. Natl. Acad. Sci. U S A 104(9), 3414–3419 (2007)
13. Smyth, G.K.: Limma: linear models for microarray data, pp. 397–420. Springer, New York (2005)
14. Tan, A.C., Naiman, D.Q., Xu, L., Winslow, R.L., Geman, D.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics 21(20), 3896–3904 (2005)

Restricted Neighborhood Search Clustering Revisited: An Evolutionary Computation Perspective

Clara Pizzuti[1] and Simona E. Rombo[2]

[1] Institute for High Performance Computing and Networking,
National Research Council of Italy, CNR-ICAR,
Via P. Bucci 41C, 87036 Rende (CS), Italy
pizzuti@icar.cnr.it
[2] Department of Mathematics, Computer Science Section
Universitá degli Studi di Palermo, Palermo, Italy
90123 Palermo, via Archirafi 34, Italy
simona.rombo@math.unipa.it

Abstract. Protein-protein interaction networks have been broadly studied in the last few years, in order to understand the behavior of proteins inside the cell. Proteins interacting with each other often share common biological functions or they participate in the same biological process. Thus, discovering protein complexes made of groups of proteins strictly related, can be useful to predict protein functions. Clustering techniques have been widely employed to detect significative biological complexes. In this paper, we integrate one of the most popular network clustering techniques, namely the Restricted Neighborhood Search Clustering (RNSC), with evolutionary computation. The two cost functions introduced by $RNSC$, besides a new one that combines them, are used by a Genetic Algorithm as fitness functions to be optimized. Experimental evaluations performed on two different groups of interactions of the budding yeast *Saccaromices cerevisiae* show that the clusters obtained by the genetic approach are more accurate than those found by $RNSC$, though this method predicts more true complexes.

1 Introduction

Proteins are the basic constituents of living beings. It has been shown that studying how proteins interact inside the cell is necessary to understand the biological processes in which they are involved [37]. Thanks to the development of advanced high-throughput technologies, many protein-protein interactions have been discovered in the last few years (see, e.g., [15,21]). The set of all the protein-protein interactions of a given organism is its *interactome*, usually modeled by an indirect graph, called *protein-protein interaction network* (PPI network), where nodes represent involved proteins and edges encode their interactions. PPI networks received much attention in the last few years [2,10,33,36] since they can be usefully exploited to study protein functions and to infer information about conservations among species.

Proteins are organized into different putative protein complexes, each performing specific tasks in the cell [12,26]. Proteins interacting with each other often participate in the same biological processes, or can be associated with specific biological functions

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 59–68, 2013.

being strongly related [35]. Indeed, cellular functions are likely to be accomplished in a modular way, meaning that a group of physically or functionally related proteins join together to accomplish a distinct function [4]. A protein complex can then be considered as a group (cluster) of proteins contributing to the same biological functions. Their detection allows the comprehension of biologically meaningful interactions and provides important knowledge about the organization of biological systems and cellular processes, giving a valuable help in understanding the behavior of organisms.

In the last few years there has been an increasing interest in studying clustering methods able to detect groups of proteins densely interconnected. Clustering approaches to PPI networks can be broadly categorized as distance-based and graph-based ones [17]. Distance-based clustering approaches apply traditional clustering techniques, such as hierarchical clustering, by employing the concept of distance between two proteins [5,25]. Graph-based clustering techniques consider the topology of the network. These techniques find the clusters by applying different strategies. One strategy searches for sub-graphs having maximum density (e.g., [23,28], by using different notions of subgraph density. Another approach partitions the graph by optimizing a cost function [14,34]. The concept of flow simulation, though applied in different ways, is exploited in [7,13]. A statistical approach to protein clustering is taken instead in [32,9]. Very few population-based stochastic search approaches have been used for developing algorithms for community detection in PPI networks (see, e.g., [18,30,31]). Surveys describing and comparing a number of methods presented in the literature can be found in [6,22,27,29,38].

In this paper we propose to embed the cost functions introduced by King et al. [14] in a genetic algorithm, in order to evaluate the capability of evolutionary computation in predicting complexes in PPI networks. Besides the *naive cost function* and *scaled cost function*, defined in [14], a new scaled function, that takes into account the connections of nodes constituting a cluster and the size of the clusters obtained, is introduced. Experimental results on two data sets of yeast protein interactions show that the genetic approaches, when compared with $RNSC$, though predict a lower number of complexes, the predicted clusters are composed of a high percentage of true positive proteins, thus a lower number of false positive occur inside them.

The paper is organized as follows. Section 2 briefly recalls the Restricted Neighborhood Search Clustering (RNSC) Algorithm. In Section 3 its evolutionary version is proposed and described in details. In Section 4 the evaluation measures exploited to validate the performances of the introduced methods are summarized. Section 5 describes experimental evaluations performed on the budding yeast *Saccaromices cerevisiae* PPI network and points out some peculiar characteristics of the evolutionary techniques proposed in this work. Finally, in Section 6 we draw our conclusive remarks.

2 Restricted Neighborhood Search Clustering Algorithm

Restricted Neighborhood Search Clustering (RNSC) is a popular method, proposed by King et al. [14], to detect complexes in protein-protein interaction networks. $RNSC$ explores the solution space of all the possible clusterings by minimizing cost functions that reflect the number of inter-cluster and intra-cluster edges. The method partitions a

network in clusters by using two cost functions. In order to formally define these two cost functions, some formalism must be introduced.

Let $G = (V, E)$ be a graph of n nodes and m edges modeling a PPI network, and $\mathcal{S} = \{S_1, \ldots, S_k\}$ a partitioning of G in k clusters. A cross-edge in a clustering is an edge whose vertices belong to different clusters. Given a node $v \in S$, let $c_s(v) = \{(v, u) \mid u \notin S\}$ denote the number of cross-edges incident with v, and $l_s(v) = \{u \in S \mid (v, u) \notin E\}$ be the number of nodes in S not connected with v.

The first function, called the *naive cost function*, is defined as:

$$C_n(G, \mathcal{S}) = \frac{1}{2} \sum_{v \in V} (c_s(v) + l_s(v)) \tag{1}$$

Thus, the naive cost function, for each node v, computes the number of *bad connections* incident with v, i.e. one that exists between v and a node not belonging to the same cluster of v ($c_s(v)$), or one that does not exist between v and another node in the same cluster as v ($l_s(v)$).

As the authors point out, $C_n(G, \mathcal{S})$ is considered naive since it does not take into account the importance of a vertex in a graph, i.e. if it belongs to either a very large cluster or a small cluster. To reflect this concept, a second function, called the *scaled cost function*, that measures the size of the area that v effects in the clustering is introduced:

$$C_s(G, \mathcal{S}) = \frac{n-1}{3} \sum_{v \in V} \frac{(c_s(v) + l_s(v))}{\mid N(v) \cup S_v \mid} \tag{2}$$

where S_v is the cluster v belongs to, and $N(v)$ is the set of neighbour nodes of v.

The algorithm begins with a random clustering, and attempts to find a best naive clustering by moving a vertex from a cluster to another one in order to minimize the naive cost function. The choice of using the naive cost function at first, is due to the necessity of having a fairly good clustering in a fast way. Then the algorithm tries to improve the obtained solution by searching for a clustering with low scaled cost function. Since the approach is greedy, the problem of getting stack at poor local minima is dealt by making diversification moves that mix up the clustering by scattering the clusters at random. Furthermore, RNSC maintains a list of tabu moves that forbid to cycle back to previously examined solutions.

3 Evolutionary RNSC

In this section we consider the cost functions described above, and reformulate them in terms of set of nodes constituting a cluster, instead of single nodes, to obtain fitness functions that will be optimized by the evolutionary approach. Furthermore, a simplification of the scaled cost function which scales the cost function with respect to the cluster size and the crossing edges of the cluster is introduced. These three objective functions will be adopted in the genetic approach and compared with $RNSC$.

Let $\mathcal{S} = \{S_1, \ldots, S_k\}$ be a partition of the graph $G = (V, E)$, modeling a PPI network, in k clusters. Let n_s and m_s denote the number of nodes and edges, respectively, of a cluster $S \in \mathcal{S}$. Then:

$$c_s = \sum_{v \in S} c_s(v)$$

is the total number of cross-edges of the nodes of S, and

$$\overline{l}_s = \sum_{v \in S} l_s(v)$$

is the number of pairs of nodes in S not connected. The naive cost function $C_n(G, \mathcal{S})$ can be rewritten as:

$$C_n(G, \mathcal{S}) = \frac{1}{2} \sum_{s \in \mathcal{S}} c_s + \overline{l}_s \tag{3}$$

As regards the scaled cost function, we must first compute the scaled cost function for each cluster $S \in \mathcal{S}$ as follows:

$$C_s(S) = \sum_{v \in S} \frac{c_s(v) + l_s(v)}{c_s(v) + n_s} \tag{4}$$

and then sum the contribution of each of them:

$$C_s(G, \mathcal{S}) = \frac{n-1}{3} \sum_{s \in \mathcal{S}} C_s(S) \tag{5}$$

A simplification of the function (5), which scales the naive cost function of each cluster in $\mathcal{S}$ with respect to its size and the crossing edges relative to it, can be obtained as follows:

$$C_{ss}(G, \mathcal{S}) = \frac{n-1}{3} \sum_{s \in \mathcal{S}} \frac{c_s + \overline{l}_s}{c_s + n_s} \tag{6}$$

Formula (6), instead of considering the influence of a single node, it normalizes the contribution of each cluster found with respect to its size and number of connections with nodes of other clusters.

The three cost functions described above can be used inside a genetic algorithm as fitness functions to minimize, in order to partition the graph G modeling a network in dense groups of proteins.

The pseudo-code of the genetic approach is reported in Figure 1. The genetic algorithm uses the locus-based adjacency representation proposed in [24], and adopted also in [30]. In this graph-based representation an individual of the population consists of n genes $g_1, \ldots, g_n$ and each gene can assume allele values j in the range $\{1, \ldots, n\}$. Genes and alleles represent nodes of the graph $G = (V, E)$ modeling a PPI network, and a value j, assigned to the ith gene, means that proteins i and j are connected and clustered together. The initialization process assigns to each each node i one of its neighbors j. The kind of crossover operator adopted is uniform crossover. Given two parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 0 from the first parent, and the genes where the vector is a 1 from the second parent, and combines the genes to form the child. The mutation operator,

Given a network $\mathcal{N}$ and the graph $\mathcal{G} = (V, E)$ modeling it, perform the following steps:

1. **create** an initial population of random individuals whose length equals the number n of nodes of G
2. **while** termination condition is not satisfied **do**
3. **decode** each individual $I = \{g_1, \ldots, g_n\}$ of the population to obtain a partitiong $S = \{S_1, \ldots, S_k\}$ of the graph G in k connected components
4. **evaluate** the fitness of the translated individuals
5. **create** a new population of individuals by applying the variation operators
6. **end while**
7. **return** the individual having the best cost function

Fig. 1. The pseudo-code of the *GA-RNSC* approach

analogously to the initialization process, randomly assigns to each node i one of its neighbors.

The algorithm, for a fixed number of generations, evolves the population of individuals, decodes each chromosome to determine the division of the graph in k connected components, computes the fitness function of each member of the population, and applies the specialized variation operators described above to produce the new population. At the end of the evolution process, the individual having the best cost function is returned as solution. It is worth to note that decoding can be efficiently performed by using a disjoint set algorithm, as described in [8].

4 Evaluation Measures

In the following we describe some validation measures widely exploited in the literature [1,3,16] that will be used for the comparative analysis presented in this work. For the generic predicted cluster P_i and the generic known complex K_j, let $\mid P_i \mid$ and $\mid K_j \mid$ be their sizes, respectively. Furthermore, let $\mid P_i \cap K_j \mid$ be the size of the intersection set of the predicted cluster and the known complex. To evaluate how a predicted cluster P_i matches a known complex K_j, the *overlapping score* between P_i and K_j is defined as

$$OS(P_i, K_j) = \frac{\mid P_i \cap K_j \mid^2}{\mid P_i \mid \cdot \mid K_j \mid} \tag{7}$$

A known complex and a predicted cluster are considered a *match* [16] if $OS(P_i, K_j) \geq \sigma_{OS}$, i.e. their overlapping score is equal to or larger than a specific threshold σ_{OS}. To estimate the performance of algorithms for detecting protein complexes w.r.t. the overlapping score, the notions of *sensitivity* and *specificity*, commonly used in information retrieval and machine learning (also known as *recall* and *precision*), as well as a cumulative measure called *f-measure* are introduced.

Sensitivity: $S_n = \frac{TP}{TP+FN}$ is the fraction of the true-positive predictions out of all the true predictions, where TP (true positive) is the number of the predicted clusters matched by the known complexes with $OS(P_i, K_j) \geq \sigma_{OS}$, and FN (false negative) is the number of the known complexes that are not matched by the predicted clusters.

Specificity: $S_p = \frac{TP}{TP+FP}$ is the fraction of the true-positive predictions out of all the positive predictions, where FP (false positive) equals the total number of the predicted clusters minus TP.

F-measure: $F_m = \frac{2 \cdot S_n \cdot S_p}{S_n + S_p}$ is a measure that summarizes sensitivity and specificity. High values of f-measure means that both sensitivity and specificity are sufficiently high.

5 Experimental Results

In this section we present the results of the genetic approaches on two PPI networks and compare them with those obtained by running $RNSC$. In the following, depending on the fitness function used, i.e. formulas (3) for naive cost function, (5) for scaled cost function, and (6) for simplified scaled cost function, we refer to the genetic algorithm as GA_n-*RNSC*, GA_s-*RNSC*, and GA_{ss}-*RNSC*, respectively. The parameters of the genetic algorithm have been fixed as follows. Population size 100, number of generations 100, elite reproduction 10% of the population size, roulette selection function, crossover 0.8, mutation 0.2. This values have been chosen by taking into account the experimental evaluation reported in [30]. The implementation has been written in MATLAB 7.14 R2012a, using Genetic Algorithms and Direct Search Toolbox 2. As regards $RNSC$ we used the optimal parameter values reported in [6].

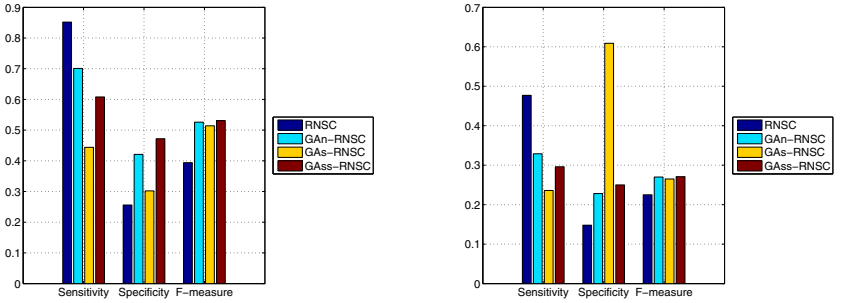
We ran the methods on two different data sets containing yeast protein interactions downloadable from http://faculty.uaeu.ac.ae/nzaki/ProRank.htm. The first dataset, denoted Yeast-D1, is that used by Gavin et al. in [11], and the second one, denoted Yeast-D2, contains yeast protein interactions generated by different experiments. Zaki et al. [39], however, filtered these two networks to delete unreliable interactions and obtained 990 proteins with $4,687$ interactions for Yeast-D1, and $1,443$ proteins with $6,993$ interactions for Yeast-D2. The reference sets of gold standard complexes include 81 (Cmplx-D1) and 162 (Cmplx-D2) hand-curated complexes from MIPS [19,20].

First of all in Table 1 the average number of complexes found by the genetic algorithms on the two yeast networks, along with the standard deviation std, are reported. The methods behave in a rather different way. $RNSC$ obtains the highest number of clusters. When the naive cost function (formula (3)) is adopted, a considerable number of clusters with smaller size with respect to the true complexes are obtained also by GA_n-*RNSC*. The opposite behavior can be observed with the scaled cost function

Table 1. Complexes found by the methods on Yeast-D1 and Yeast-D2 with 81 and 162 gold standard complexes, respectively

METHOD	YEAST-D1		YEAST-D2	
	NUMBER	STD	NUMBER	STD
RNSC	293	0	427	0
GA_n-*RNSC*	138.4	12.1	207.6	10.9
GA_s-*RNSC*	58.2	3.5	112.6	3.7
GA_{ss}-*RNSC*	107.8	3.2	171	3.6

(formula (5)) that induces a much lower number of clusters having larger size. With the simplified scaled cost function (formula (6)), GA_{ss}-*RNSC* produces a number of clusters higher than GA_s-*RNSC*, and lower than GA_n-*RNSC*. These numbers differ from the true number of complexes and suggest that $RNSC$ divides the complexes in small groups of proteins, GA_n-*RNSC* has a similar but less emphasized behavior, GA_s-*RNSC*, on the contrary, joins complexes, while GA_{ss}-*RNSC* also splits complexes, but for a lower percentage of groups than GA_n-*RNSC*. Thus the optimization of the cost functions of $RNSC$ through evolutionary computation produces predicted clusters that are sensibly dissimilar from those generated by $RNSC$.

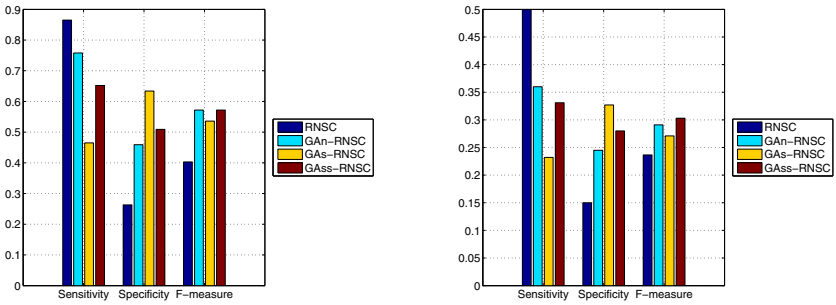


(a) Yeast-D1 (b) Yeast-D2

Fig. 2. Sensitivity, specificity, and f-measure values for (a) Yeast-D1 and (b) Yeast-D2 networks with overlapping score $OS(P_i, K_j) \geq 0.2$

Figure 2 shows sensitivity, specificity, and f-measure values obtained by the genetic approaches and $RNSC$ when the overlapping score $OS(P_i, K_j) \geq 0.2$. The first observation is that $RNSC$ has a higher sensitivity value compared with the genetic algorithms on both the two networks. This means that $RNSC$ is able to predict a higher number of complexes, out of all the true complexes. This result can be explained by the high number of clusters that $RNSC$ finds. It is worth to note that, the definition of overlapping score (formula (7)) penalizes those methods that obtain clusters with size $\mid P_i \mid$ much greater than the true complex size $\mid K_i \mid$. In fact the denumerator of (7) has a higher value if the cluster size $\mid P_i \mid$ is high, and, consequently, $OS(P_i, K_j)$ is lower. This bias can be observed also for the three evolutionary methods. GA_n-*RNSC*, GA_{ss}-*RNSC*, and GA_s-*RNSC* present a decreasing number of predicted clusters, and thus the predicted clusters are of increasing size. The figure shows that sensitivity values reflect the size of the predicted clusters. The lower the size, the higher the corresponding sensitivity values.

On the other hand, from the figure we can observe that specificity and f-measure are both higher for the genetic approaches. Higher specificity means that the predicted clusters have a high percentage of proteins effectively belonging to the true complex, thus the fraction of false positive is low. In particular, GA_s-*RNSC* is the best performing on Yeast-D2, while GA_{ss}-*RNSC* reaches better values of specificity on Yeast-D1.



(a) Yeast-D1 (b) Yeast-D2

Fig. 3. Sensitivity, specificity, and f-measure values for (a) Yeast-D1 and (b) Yeast D2 networks when overlapping score $OS_J(P_i, K_j) \geq 0.2$

In order to more deeply investigate the effects of the overlapping score $OS(P_i, K_j)$, we considered a different definition of overlapping score based on the Jaccard coefficient, that is:

$$OS_J(P_i, K_j) = \frac{\mid P_i \cap K_j \mid}{\mid P_i \cup K_j \mid} \tag{8}$$

Sensitivity, specificity and f-measure have been recomputed and the values obtained when the overlapping score $OS_J(P_i, K_j) \geq 0.2$ are reported in Figure 3. Also in this experiment it is possible to observe that sensitivity values obtained by $RNSC$ are higher. However, specificity and f-measure are better for all the three fitness functions used, confirming the above observations.

From the described experimental campaign, we can conclude that evolutionary computation allows to improve specificity w.r.t. the $RNSC$ method, still retaining good values of sensitivity. In particular, $RNSC$ returns in output many clusters, and each of them only partially overlaps with some true complexes. On the contrary, GA-$RNSC$ approaches predict a lower number of clusters, but their overlapping with true complexes is larger. As an example, GA_n-*RNSC* correctly found a complex of Yeast-D1 (20 of 22 proteins) recognized to be a RNA polymerase II holoenzyme/mediator subunit, while GA_s-*RNSC* was able to find a full complex in Yeast-D2 made of cAMP-dependent protein kinases.

6 Conclusions

In this work we showed the capability of evolutionary computation to predict complexes in PPI networks by embedding the cost functions introduced by King et al. [14] in a genetic algorithm. A new scaled function able to take into account, besides the connections of nodes constituting a cluster, also the size of the clusters obtained, is also introduced. Experimental results on two data sets of yeast protein interactions proved that the genetic approaches, when compared with $RNSC$, return complexes with a

higher percentage of true positive proteins. Future work aims to improve the evolutionary approach by considering different combinations of the fitness functions, possibly enriched with local search strategies.

Acknowledgements. This work has been partially supported by the project *MERIT : MEdical Research in Italy*, funded by MIUR.

References

1. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. BMC Bioinformatics 7(207) (2006)
2. Atias, N., Sharan, R.: Comparative analysis of protein networks: hard problems, practical solutions. Commun. ACM 55(5), 88–97 (2012)
3. Bader, G., Hogue, H.: An automated method for finding molecular complexes in large protein-protein interaction networks. BMC Bioinformatics 4(2) (2003)
4. Barabási, A., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Review Genetics 5, 101–113 (2004)
5. Blatt, M., Wiseman, S., Domany, E.: Superparamagnetic clustering of data. Phisical Review Letters 76(18), 3251–3254 (1996)
6. Brohèe, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7, 488 (2006)
7. Cho, Y.-R., Hwang, W., Ramanathan, M., Zhang, A.: Semantic integration to identify overlapping functional modules in protein interaction networks. BMC Bioinformatics 8, 265 (2007)
8. Thomas, H., Cormen, C.E., Leiserson, R.L.: Rivest, and Clifford Stein. In: Introduction to Algorithms, 2nd edn. MIT Press (2007)
9. Farutin, V., Robinson, K., Lightcap, E., Dancik, V., Ruttenberg, A., Letovsky, S., Pradines, J.: Edge-count probabilities for the identification of local protein communities and their organization. Proteins: Structure, Function, and Bioinformatics 62, 800–818 (2006)
10. Ferraro, N., Palopoli, L., Panni, S., Rombo, S.E.: Asymmetric comparison and querying of biological networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8, 876–889 (2011)
11. Gavin, A.C., et al.: Proteome survey reveals modularity of the yeast cell machinery. Nature 440, 631–636 (2006)
12. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: Clustering algorithm based graph connectivity. Nature 402, 47–52 (1999)
13. Hwang, W., Cho, Y.-R., Zhang, A., Ramanathan, M.: A novel functional module detection algorithm for protein-protein interaction networks. Algorithms for Molecular Biology 1(24) (2006)
14. King, A.D., Przulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20(17), 3013–3020 (2004)
15. Krogan, N.J., et al.: Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature 440(7084), 637–643 (2006)
16. Li, M., Chen, J., Wang, J., Hu, B., Chen, G.: Modifying the DPClus algorithm for identifying protein complexes based on new topological structures. BMC Bioinformatics 9 (2008)
17. Lin, C., Cho, Y., Hwang, W., Pei, P., Zhang, A.: Clustering methods in protein-protein interaction network. Knowledge Discovery in Bioinformatics: Techniques, Methods and Application. John Wiley & Sons, Inc. (2006)
18. Liu, H., Liu, J.: Clustering protein interaction data through chaotic genetic algorithm. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 858–864. Springer, Heidelberg (2006)

19. Mewes, H.W., et al.: MIPS: a database for genomes and protein sequences. Nucleic Acids Res. 30(1), 31–34 (2002)
20. Mewes, H.W., et al: MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res. 34(database issue 1), 169–172 (2006)
21. Miller, J.P., et al.: Large-scale identification of yeast integral membrane protein interactions. Proc. Natl. Acad. Sci. USA 102(34), 12123–12128 (2005)
22. Moschopoulos, C.N., Pavlopoulos, P.A., Iacucci, E., Aerts, J., Likothanassis, S., Schneider, R., Kossida, S.: Which clustering algorithm is better for predicting protein complexes? BMC Research Notes 4(549) (2011)
23. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
24. Park, Y.J., Song, M.S.: A genetic algorithm for clustering problems. In: Proc. of 3rd Annual Conference on Genetic Algorithms, pp. 2–9 (1989)
25. Pei, P., Zhang, A.: A two-step approach for clustering proteins based on protein interaction profiles. In: IEEE Int. Symposium on Bioinformatics and Bioengeneering (BIBE 2005), pp. 201–209 (2005)
26. Pereira, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. Proteins: Structure, Fuctions, and Bioinformatics (20), 49–57 (2004)
27. Pizzuti, C., Rombo, S.E.: Discovering Protein Complexes in Protein Interaction Networks in Biological Data Mining in Protein Interaction Networks. In: Li, X.-L., Ng, S.-K. (eds.). IGI Global- Medical Inf. Science Ref. (2009)
28. Pizzuti, C., Rombo, S.E.: A coclustering approach for mining large protein-protein interaction networks. IEEE/ACM Trans. Comput. Biology Bioinform. 9(3), 717–730 (2012)
29. Pizzuti, C., Rombo, S.E., Marchiori, E.: Complex detection in protein-protein interaction networks: A compact overview for researchers and practitioners. In: Giacobini, M., Vanneschi, L., Bush, W.S. (eds.) EvoBIO 2012. LNCS, vol. 7246, pp. 211–223. Springer, Heidelberg (2012)
30. Pizzuti, C., Rombo, S.E.: Experimental evaluation of topological-based fitness functions to detect complexes in PPI networks. In: Proc. of the Genetic and Evolutionary Computation Conference (Gecco 2012), pp. 193–200 (2012)
31. Ravaee, H., Masoudi-Nejad, A., Omidi, S., Moeini, A.: Improved immune genetic algorithm for clustering protein-protein interaction network. In: Proceedings of the 2010 IEEE International Conference on Bioinformatics and Bioengineering, BIBE 2010, pp. 174–179. IEEE Computer Society (2010)
32. Samantha, M.P., Liang, S.: Predicting protein functions from redundancies in large-scale protein interaction networks. Proc. of the National Academy of Science 100(22), 12579–12583 (2003)
33. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Molecular Systems Biology 3(88) (2007)
34. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. PNAS 100, 12123–12128 (2003)
35. Tornw, S., Mewes, H.W.: Functional modules by relating protein interaction networks and gene expression. Nucleic Acids Research 31(21), 6283–6289 (2003)
36. De Virgilio, R., Rombo, S.E.: Approximate matching over biological RDF graphs. In: Proceedings of the ACM Symposium on Applied Computing, SAC 2012, pp. 1413–1414 (2012)
37. von Mering, D., Krause, C., et al.: Comparative assessment of a large-scale data sets of protein-protein interactions. Nature 31, 399–403 (2002)
38. Wang, J., Li, M., Deng, Y., Pan, Y.: Recent advances in clustering methods for protein interaction networks. BMC Genomics 11(suppl. 3), S10 (2010)
39. Zaki, N., Berengueres, J., Efimov, D.: Prorank: a method for detecting protein complexes. In: Proc. of the Genetic and Evolutionary Computation Conference (Gecco 2012), pp. 209–216 (2012)

Class Dependent Feature Weighting and K-Nearest Neighbor Classification

Elena Marchiori

Institute for Computing and Information Sciences, Radboud University Nijmegen,
The Netherlands
elenam@cs.ru.nl

Abstract. Feature weighting in supervised learning concerns the development of methods for quantifying the capability of features to discriminate instances from different classes. A popular method for this task, called RELIEF, generates a feature weight vector from a given training set, one weight for each feature. This is achieved by maximizing in a greedy way the sample margin defined on the nearest neighbor classifier. The contribution from each class to the sample margin maximization defines a set of *class dependent feature weight* vectors, one for each class. This provides a tool to unravel interesting properties of features relevant to a single class of interest.

In this paper we analyze such class dependent feature weight vectors. For instance, we show that in a machine learning dataset describing instances of recurrence and non-recurrence events in breast cancer, the features have different relevance in the two types of events, with size of the tumor estimated to be highly relevant in the recurrence class but not in the non-recurrence one. Furthermore, results of experiments show that a high correlation between feature weights of one class and those generated by RELIEF corresponds to an easier classification task.

In general, results of this investigation indicate that class dependent feature weights are useful to unravel interesting properties of features with respect to a class of interest, and they provide information on the relative difficulty of classification tasks.

1 Introduction

Feature selection is a central problem in machine learning, because of its use in a wide range of real-life applications, such as in pattern recognition, text categorization and biological and biomedical data analysis [6]. Feature selection in supervised learning is motivated by the fact that using all available features may negatively affect generalization performance, especially in the presence of irrelevant or redundant features. Therefore, feature selection aims at making good predictions with few features.

Many feature selection algorithms have been proposed (see for instance the overviews in [6,10]). In particular, feature weighting assigns real values (instead of zero or one) to features, describing their relevance to a learning problem. Among the existing feature weighting algorithms, RELIEF [7,9,8] is considered one of the most successful methods, due to its simplicity and effectiveness [1]. Interesting formalizations of RELIEF

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 69–78, 2013.

have been proposed and used for developing new feature weighting algorithms [5,11]. In particular it was shown that RELIEF can be interpreted as an online solution to a convex optimization problem, which maximizes a margin-based objective function, where the margin is defined on the 1-nearest neighbor (1-NN) classifier. This explains its good performance both when compared with filter methods, due to the performance feedback of a nonlinear classifier when searching for useful features, and when compared with wrapper methods, since it optimizes a convex problem, hence avoids any exhaustive or heuristic combinatorial search and can be implemented very efficiently [11].

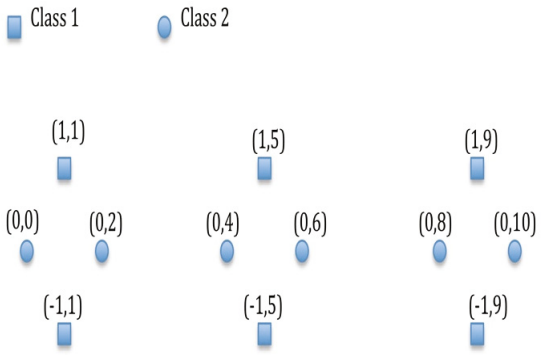


Fig. 1. Toy example

This paper investigates a decomposition of RELIEF into class dependent feature weight terms, one for each class of the learning problem. Each class dependent term induces a weighted distance enlarging the sample margin of the corresponding class. This approach can be viewed as the supervised counterpart of clustering in subspaces spanned by different combinations of dimensions via local weightings of features (see, e.g., [4,2]).

We show that complementary characteristics of a feature in different classes may yield different weight contributions which, when added, neutralize each other. This may prevent to detect the relevance of some features for a single class.

This situation is illustrated by the following toy example. RELIEF applied to the training data shown in Figure 1 assigns zero weight to all the features. However, if the class dependent feature weight terms are considered, the two features become relevant for class 1 and 2, respectively. This is shown in detail in Section 2.1.

The goal of this paper is to analyze class dependent feature weight vectors. First, we investigate whether class dependent feature weights provide useful information about the relative difficulty of classification tasks. Second, we investigate whether class dependent feature weights provide better information about the relevance of features than RELIEF for the underlying phenomenon under study.

Results of experiments show that a high correlation between feature weights of one class and those generated by RELIEF are associated to an easier classification task.

Furthermore, we show that in a machine learning dataset describing instances of recurrence and non-recurrence events in breast cancer, the features have different relevance in the two types of events. In particular, the size of the tumor is estimated to be highly relevant in the recurrence class but not in the non-recurrence one.

In general, results of this investigation indicate that class dependent feature weights are useful to unravel interesting properties of features with respect to a class of interest, and can be used to analyze comparatively the difficulty of classification tasks.

2 Methods

We use the variant of RELIEF acting on all the training set instances [8] (see Algorithm 1 for binary classification problems).

Algorithm 1. RELIEF

Require: X, l: training data, each instance vector x has m features and class $l(x)$
Ensure: w: feature weight vector
  w = vector with m zeros
  for x in X **do**
    $w = w + \sum_{z \in \mathtt{KNN}(x,c), c \neq l(x)} |x - z| - \sum_{z \in \mathtt{KNN}(x,l(x))} |x - z|$;
  end for

X denotes a dataset of n instances, and x, z generic instances. Each instance $x = (x_1, \ldots, x_m)$ is a real-valued vector of dimension m, whose entries are here called features.

C denotes the set of class labels; $l : X \to C$ is the function mapping each instance x to its class label $l(x)$. Let c be a generic element of C, and X_c be the subset of X consisting of those instances having class label equal to c. Then $\mathtt{KNN}(x, c)$ is the set of K nearest neighbors (with respect to the Euclidean distance) of x computed using only the instances in X_c, excluded x.

For two instance vectors x and y, $|x-y|$ denotes the vector consisting of the absolute value of the difference of each pair of corresponding entries in x and y. For instance, if $x = (1, 2)$ and $y = (3, 4)$ the $|x - y| = (2, 2)$.

Given a training set of labeled instances, RELIEF generates the feature weight vector w of size equal to the number (m) of features. Each feature's weight in w is initialized to zero and updated iteratively by processing each instance vector x of X as follows. The K nearest neighbors of x from the opposite class are computed, and for each one of them, say z, the vector $|x - z|$ is added to w. The K nearest neighbors from the same class are computed, and for each one of them, say z, the vector $|x - z|$ is subtracted from w.

2.1 Class Dependent Feature Weighting

The feature weight vector computed by RELIEF can be re-written as $w = \sum_{c \in C} w_c$ where

$$w_c = \sum_{x \in X_c} \left\{ \sum_{z \in \mathtt{KNN}(x,c)} -|x - z| + \sum_{\substack{z \in \mathtt{KNN}(x,c') \\ c' \neq c}} |x - z| \right\}.$$

The term w_c can be viewed as a feature weight vector conditioned to class c: we call it *class dependent feature weight* vector.

For the toy example described in the Introduction (see Figure 1) the class dependent feature weight vectors for class 1 and 2 are $(-6, 6)$ and $(6, -6)$, respectively. Their sum is the vector $(0, 0)$. Therefore RELIEF assigns weight zero to all features. However, the two class dependent weight vectors estimate the second and first feature as 'relevant' for class 1 and 2, respectively.

This observation shows that, by averaging the weigths of features across different classes, information about the relevance of features with respect to a single class can be lost, while this information is visible if the weigths of features for each class are kept separated.

A class dependent feature weight w_c has a direct interpretation in terms of sample margin [5,11]. In fact w_c can be viewed as the result of a greedy procedure for maximizing the sample margin *conditioned to class c* as described by the following objective function:

$$\theta_c = \sum_{\substack{x \ s.t. \\ l(x)=c}} \left(\sum_{c' \neq c} \sum_{z \in \mathtt{KNN}_w(x,c')} \|x - z\|_w - \right.$$

$$\left. \sum_{z \in \mathtt{KNN}_w(x,c)} \|x - z\|_w \right).$$

Here $\|x - z\|_w$ denotes the weighted Euclidean distance between x and z with respect to feature weights $w = w_1, \ldots, w_m$, $w_i \geq 0$, defined as $\left(\sum_{i=1}^{m} w_i (x_i - z_i)^2\right)^{-1/2}$. $\mathtt{KNN}_w(x)$ represents the list of K nearest neighbors of x in the training set computed using the weighted Euclidean distance $\| \cdot \|_w$.

Instead, the greedy procedure of RELIEF maximizes the sample margin over the entire training set, and in this way it considers the sum of the possibly competing objectives θ_c's.

3 Applications

The goals of our investigation are twofold. First, we investigate whether class dependent feature weights provide useful information about the difficulty of the underlying classification task. Second, we investigate whether class dependent feature weights provide better information than RELIEF does about the relevance of features in relation to the underlying phenomenon under study.

To this aim we consider a number of life sciences datasets available at the UCI Machine Learning repository (see http://archive.ics.uci.edu/ml/datasets.html). Their characteristics are given in Table 1.

Table 1. Datasets used in the experiments. CL = number of classes, TR = training set, TE = test set, FS = number of features, Cl.Inst. = number of instances in each class.

DATASET	FS	TR	CL.INST.	TE	CL.INST.
B.CANCER	9	200	140-60	77	56-21
DIABETES	8	468	300-168	300	200-100
HEART	13	170	93-77	100	57-43
SPLICE	60	1000	525-475	2175	1123-1052
THYROID	5	140	97-43	75	53-22
BREAST-W	9	546	353-193	137	91-46
BUPA	6	276	119-157	69	26-43
PIMA	8	615	398-217	153	102-51

Examples of class dependent and RELIEF weights of the considered problems are shown in Figure 4. The plots in this figure show that in some cases the two classes are in strong disagreement with respect to the way they estimate the relevance of different features.

For instance, on the B.Cancer data the feature rankings induced by the two class dependent weight vectors are rather different. A similar situation can be observed for the Bupa data, where the resulting rankings are the reverse of each other. These observations show that also on real-life problems the contributions of the classes to the feature weight vector computed by RELIEF may be rather different.

3.1 Class Dependent Weights and Relative Difficulty of the Classification Task

We want to assess whether the *diversity* between the class dependent feature weights and the weights computed by RELIEF and the *hardness* of classification tasks are linked with each other.

To this end we perform 10 runs on each dataset. In each run a partition of the dataset into training and test set is used, where the size of training and test set reported in the UCI ML repository are employed (see Table 1). On each partition we compute the class dependent weight vectors and the RELIEF weights using the training data, and apply the K-NN classifier (with K=5) to the test data.

We measure hardness using the mean test accuracy, and diversity using the mean of the maximum linear correlations between class dependent feature weights and the RELIEF ones (see Table 2, columns 'Corr1', 'Corr2' , and 'Max Corr').

Then we use the Spearman's correlation to assess how well the relationship between these two variables can be described using a monotonic function. A positive Spearman correlation coefficient corresponds to an increasing monotonic trend between the two variables.

We compute the Spearman's correlation coeffient ρ with right tail (that is, with alternative hypothesis 'correlation is greater than zero' against which to compute p-values for testing the hypothesis of no correlation) between the variables 'hardness' and 'diversity' as described by the vectors generated by computing their values on each dataset. The resulting hardness and diversity vectors are the columns 'Acc' and 'Max Corr' of

Table 2. Results for datasets used in the experiments. Acc = average test accuracy of K-NN classifier (with K=5) over 10 runs, Ave Corr1 = average of the correlations between RELIEF weights and class 1 dependent weights, Ave Corr2 = average of the correlations between RELIEF weights and class 2 dependent weights. Max Corr = average over 10 runs of the maximum between Corr 1 and Corr 2. Standard deviation (std) over 10 runs is reported between brackets.

DATASET	ACC (STD)	CORR1 (STD)	CORR2 (STD)	MAX CORR (STD)
B.CANCER	70.12 (5.44)	-0.35 (15.75)	64.43 (10.79)	64.43 (10.79)
DIABETES	70.65 (4.57)	-5.73 (24.38)	86.24 (4.85)	86.24 (4.85)
HEART	80.00 (3.39)	36.33 (16.64)	63.88 (16.37)	67.41 (10.38)
SPLICE	72.89 (1.72)	43.50 (13.65)	91.64 (3.37)	91.64 (3.37)
THYROID	91.86 (3.84)	42.90 (24.65)	89.03 (3.67)	89.03 (3.67)
BREAST-W	97.95 (1.32)	21.93 (19.29)	97.91 (1.01)	97.91 (1.01)
BUPA	65.07 (6.12)	61.29 (34.41)	-13.47 (38.37)	67.53 (23.31)
PIMA	72.73 (1.08)	24.45 (35.18)	89.53 (6.61)	89.53 (6.61)

Table 2. Computation of the coefficient yields a positive correlation $\rho = 0.5952$ with p-value 0.06.

If we consider the list of test accuracies and the list of maximum correlations (between class dependent feature weights and the RELIEF ones) generated in the 10 runs instead of their mean, then a smaller yet more significant correlation is obtained, $\rho = 0.4155$ with p-value 0.0001.

Furthermore, if we consider the test classification achieved using only the top 5 features selected using the feature ranking induced by the weights generated by RELIEF, then the correlation signal between hardness and diversity becomes highly significant. Indeed, the Spearman's correlation coefficients become $\rho = 0.8333$ (p-value 0.0154) and $\rho = 0.6476$ (p-value 0) when the mean (over the 10 runs) results and the results in all the runs are considered, respectively.

The above results indicate that class dependent feature weights provide information about the relative difficulty of the learning tasks (as measured by the test error).

3.2 Class Dependent Weights and Feature Relevance

In this section we analyze the class dependent feature weights and the RELIEF ones in the context of breast cancer data analysis, using two breast cancer datasets B.Cancer and the Breast-W.

The B.Cancer Data. This dataset was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. It consists of instances from recurrence (class 2) and non-recurrence (class 1) breast cancer patients. The instances consist of the following features:

1. age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99,
2. menopause: lt40, ge40, premeno,
3. tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59,
4. inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39,
5. node-caps: yes, no,
6. deg-malig: 1, 2, 3,
7. breast: left, right,
8. breast-quad: left-up, left-low, right-up, right-low, central,
9. irradiat: yes, no.

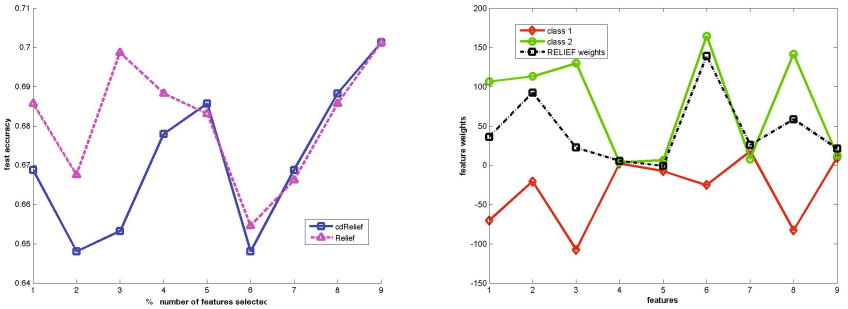


Fig. 2. Average accuracy of the K-NN classifier (K=5) for different number of selected features (left) and example of feature weight vectors (right) on dataset B.Cancer

Figure 2 (right side) shows the weights of the features according to RELIEF, and the class dependent ones. The two classes assign rather different values to some features. In particular, class 2 'recurrence events' assigns high values to feature 'tumor-size' (feature 3), 'deg-malig' (6), and breast-quad (8), while class 1 'non-recurrence-events' considers them not relevant (assigns negative weights).

Thus when the class dependent relevance values are added by RELIEF, feature 'tumor-size' (3) becomes not very relevant for this learning task, less relevant than feature 1 ('age') and 2 ('menopause'). However, 'tumor-size' is recognized as relevant for the 'recurrence events' class. Indeed, research in breast cancer has shown tumor size to be strongly associated to the its recurrence hence to survival (see e.g. [3]).

Figure 2 (left side) shows the average test accuracy (over 10 runs) obtained by the K-NN classifier (with K=5) when varying the number of selected features. The features are selected using the ranking induced either by the weights computed by RELIEF or by the class dependent weights with highest variance ('cdRelief' classifier in Figure 2). Results show that test accuracy decreases when using only one class to estimate the relevance of features. This is expected since the feature ranking is biased towards one class and the two classes have different feature rankings.

Wisconsin Breast Cancer Dataset. The Wisconsin Breast Cancer dataset (Original), donated from Dr. WIlliam H. Wolberg (University of Wisconsin Hospitals Madison, Wisconsin, USA), is publicly available at the UCI Machine Learning Repository. The dataset contains malignant (cancerous, class 2) and benign (non-cancerous, class 1) instances. The instances consist of the following features:

1. Clump Thickness: 1 - 10,
2. Uniformity of Cell Size: 1 - 10,
3. Uniformity of Cell Shape: 1 - 10,
4. Marginal Adhesion: 1 - 10,
5. Single Epithelial Cell Size: 1 - 10,
6. Bare Nuclei: 1 - 10,
7. Bland Chromatin: 1 - 10,
8. Normal Nucleoli: 1 - 10,
9. Mitoses: 1 - 10.

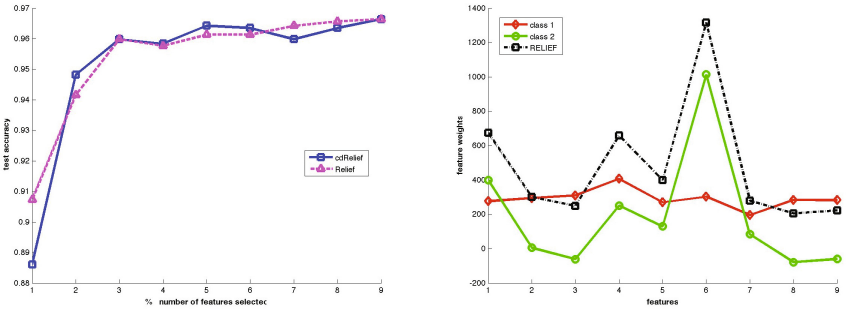


Fig. 3. Average accuracy of the K-NN classifier (K=5) for different number of selected features (left) and example of feature weight vectors (right) on dataset Breast-W

On this dataset the class dependent feature weights for class 2 (malignant cancer) are in strong accordance with those of RELIEF (see the right plot in Figure 3 for an example of weights generated by a run). Therefore the relevance as estimated by RELIEF reflects the importance of the features for malignant cancer.

Moreover the average test classification performance is not significantly affected when using the feature ranking generated by class 2 (see the performance of 'cdRelief' in the left plot in Figure 3). Indeed, on this problem both the average accuracy (97.95) and the maximum Pearson correlation, that is, that between RELIEF and class 2 weights (97.91) are high.

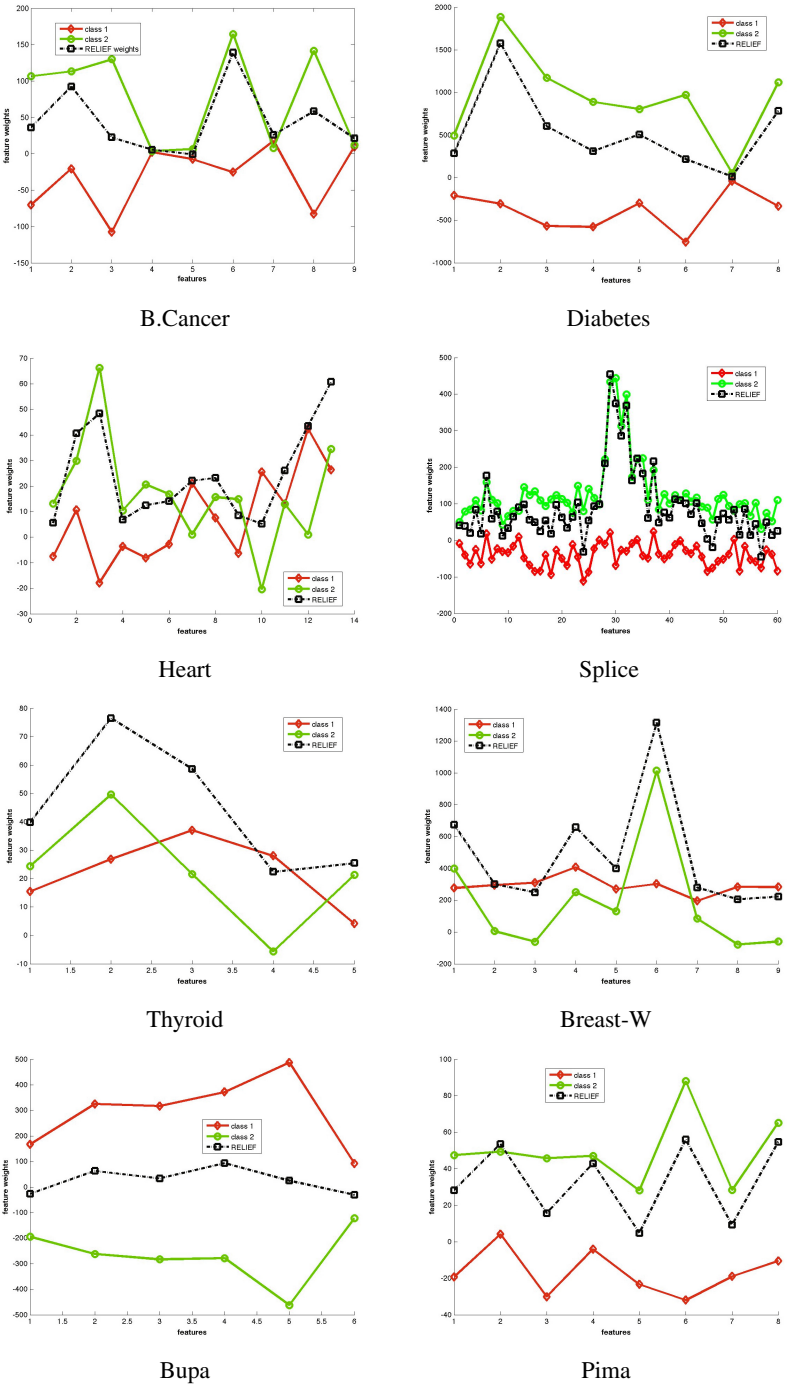


Fig. 4. Examples of class dependent and RELIEF weights of the considered datasets

4 Conclusions

We investigated a decomposition of RELIEF into class dependent feature weight vectors, where each vector describes the relevance of features conditioned to one class. The results of experiments indicated the usefulness of this decomposition for unraveling relevant features for a single class and for providing information about the difficulty of the considered learning task.

In general, results indicated that using only one class to estimate the relevance of features is not beneficial for the classification performance. This is expected, and shows that feature relevance for classification is different than feature relevance for a single class. The latter type of relevance is important when the goal is to unravel useful information about the phenomenon under study, like in the example about the recurrence of breast cancer events we discussed in this paper.

The contributions of this work provide initial insights and results about the advantages of using feature relevance in a way that depends on the single classes. Future work includes the development of a theoretical and methodological framework in order to better understand, use and combine class dependent feature weights. For instance, on the methodological side, an interesting problem for future research is to investigate whether the use of multi-objective optimization for feature weighting could be employed to improve also the classification performance, where the objectives are the θ_c's (that is, the sample margin conditioned to the classes c in C).

References

1. Dietterich, T.G.: Machine-learning research: Four current directions. The AI Magazine 18(4), 97–136 (1998)
2. Domeniconi, C., Gunopulos, D.: andS. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. Locally adaptive metrics for clustering high dimensional data. Data Min. Knowl. Discov. 14(1), 63–97 (2007)
3. Elkin, E.B., Hudis, C., Begg, C.B., Schrag, D.: The effect of changes in tumor size on breast carcinoma survival in the u.s.: 1975-1999. Cancer 104(6), 1149–1157 (2005)
4. Friedman, J.H., Meulman, J.J.: Clustering objects on subsets of attributes. Journal of the Royal Statistical Society 6, 815–849 (2004)
5. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection - theory and algorithms. In: ICML (2004)
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
7. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: National Conference on Artificial Intelligence, pp. 129–134 (1992)
8. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence Journal 97(1-2), 273–324 (1997)
9. Kononenko, I.: Estimating attributes: Analysis and extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
10. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC (2007)
11. Sun, Y.: Iterative RELIEF for feature weighting: Algorithms, theories, and applications. IEEE TPAMI 29(6), 1035–1051 (2007)

Simultaneous Sample and Gene Selection Using T-score and Approximate Support Vectors

Piyushkumar A. Mundra[1], Jagath C. Rajapakse[1,2,3], and D.A.K. Maduranga[1]

[1] Bioinformatics Research Center, School of Computer Engineering, Nanyang Technological University, Singapore
[2] Singapore-MIT Alliance, Singapore
[3] Department of Biological Engineering, Massachusetts Institute of Technology, USA
asjagath@ntu.edu.sg

Abstract. T-score, based on *t*-statistics between samples and disease classes, is a widely used filter criterion for gene selection from microarray data. However, classical T-score uses all the training samples but for both biological and computational reasons, selection of relevant samples for training is an important step in classification. Using a modified logistic regression approach, we propose a sample selection criterion based on T-score and develop a backward elimination approach for gene selection. The method is more stable and computationally less costly compared to support vector machine recursive feature elimination (SVM-RFE) methods.

Keywords: data point selection, gene selection, instance selection, logistic regression.

1 Introduction

Gene selection is a vital step in the analysis of microarray gene-expression data and several approaches have been proposed earlier [1–8]. The methods of gene selection can be broadly categorized into filter, wrapper, or embedded methods. Filter methods are simple and computationally efficient, but have lower performance than the other methods. T-score based on *t*-statistics measuring correlation between input features and output class labels is commonly used as filter criterion for sample classification [1]. Other popular filter methods include Relief [9], correlation based feature selection [10], minimum redundancy maximum relevancy [11]. For more details on filter methods, readers are referred to [2]. However, in classical filter approaches, all the training samples are used for gene ranking while ignoring the relevance and quality of data samples.

On the other hand, popular wrapper and embedded methods include Support Vector Machine Recursive Feature Elimination [5] and its variants [3, 12–16], random forest-RFE [17], elastic net [18] etc. All these methods predominantly use classifier performance in ranking genes. Many classifiers, such as support vector machines (SVM), boosting algorithms, and logistic regression etc. indicate

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 79–90, 2013.

that all samples in a training data may not be equally relevant for the classification task [19, 20]. Removal of samples (or data points) that do not provide useful information for classification improves the performance. In microarray analysis, due to heterogeneity of tissues and cell assays, the datasets are inherently multimodal [21] and therefore qualities of samples vary. Using the classical theory of margin of classifier [19], sample points could be classified into three types: within the margin, on the margin, and away from the margin. For a classification task, various theories, including SVM and boosting techniques, suggest that the points on the margin and within the margin are more important than the samples away from the margin. Giving more importance to samples on or within the margin boundary may reduce the error variance in feature selection [22]. Earlier, the importance of selection of sample in active feature selection and dimensionality reduction was demonstrated using *kd*-tree algorithm [23, 24]. A genetic algorithm/k-nearest neighbour based approach was proposed for simultaneous selection of samples and metabolomic features [6]. Similarly, a modified particle swarm algorithm was combined with SVM for simultaneous sample selection and gene ranking [25]. Very recently, sample weighting based gene selection algorithm was proposed where sample weights are determined according to its influence to the estimation of feature relevance [26].

Along with better classification, a method of identification of true markers should be reproducible (stable) with respect to variations of the samples [16, 27]. Instability of a gene ranking casts doubts over computational results and hence does not give confidence for further biological validation. Stability of a gene selection method depends on many factors which includes sample size, treatment to correlative structure and underlying data distribution. However, an improvement in stability should not decrease the accuracy of sample.

Recently, predictive performance, stability and functional interpretability of 32 gene selection methods were analysed on 4 breast cancer datasets and results indicate that a simple Student's t-test (similar to T-score) performs the best [28]. However, the issue of relevant samples still persists. In our previous work, we decomposed T-score into two parts corresponding to relevant samples and non-relevant samples to show the importance of sample selection in T-score. And thereby a support vector based t-score recursive feature elimination (SVt-RFE) algorithm was proposed for feature selection [29, 30]. However, this algorithm uses SVM to select the samples and hence is computationally expensive. It also suffers from low stability. In this paper, we propose a gene selection method to improve stability and computational complexity of SVt-RFE and SVM-RFE methods without compromising on the performance of classification. To do so, we propose an efficient sample selection criterion to identify relevant samples by incorporating a modified logistic regression model, similar to SVM, using T-score as the selection criterion. A backward elimination approach is then proposed to iteratively select the relevant genes and achieve better classification accuracy than the existing methods. Our analysis indicates that the proposed algorithm improves the stability compared to SVM based approaches.

2 Method

Suppose $D = \{x_{ij}\}_{i=1,j=1}^{n,m}$ denotes a microarray dataset of m gene expression samples obtained on n genes where x_{ij} is the expression of gene i gathered in sample j. The vector $x_j = (x_{ij})_{i=1}^{n}$ denotes gene expressions on sample j and $x_i = (x_{ij})_{j=1}^{m}$ denotes the expressions of gene i across all the samples. Let two-class classification of sample j be $y_j \in \ell = \{+1, -1\}$ taking values +1 or -1 for cancerous (+ class) or benign (- class), respectively.

2.1 T-score

T-score is a ranking measure based on t-statistic between gene expressions and class labels. For gene i, T-score (r_i) is given by

$$r_i = \frac{\left|\mu_i^+ - \mu_i^-\right|}{\sqrt{\frac{m^+(\sigma_i^+)^2 + m^-(\sigma_i^-)^2}{m^+ + m^-}}} \tag{1}$$

where superscript + and − denotes positive and negative classes, respectively. The m^+, μ_i^+ and σ_i^+ represents the number of samples, the mean and standard deviation of expression values of gene i in samples of the positive class respectively. Similarly, m^-, μ_i^- and σ_i^- are defined for negative class. Higher the ranking value, more important the gene for separation of the classes is [1].

T-score is an easy and fast measure to compute as it assumes independence among genes and normality of data. However, many a times this method gives a stable gene subset which performs poor in classification compared to wrapper and embedded methods because it does not take into account the characteristics of the classifier in the ranking of genes. One way to improve the performance of this criterion is to select relevant samples when computing the T-score [29].

2.2 Efficient Sample Selection Technique

The margin of separation of SVM is defined by the support vectors or the samples on the margin. The support vectors are the samples that in fact define the discriminant function. Use of only the support vectors for gene selection was earlier demonstrated in support vector machine recursive feature elimination (SVM-RFE) method [29]. In this section, an efficient method to select samples (approximate support vectors) is proposed for gene selection. *Relevant samples* refers to those on and within the margin of separation. Using SVM, determining the margin of separation in two-class sample classification has a computational complexity of $O(\max(n, m)m^2)$. This becomes even more costly for SVM-RFE as each iteration needs retraining the SVM. Therefore, there is a need for a simpler model selecting samples on and within the margins, which is computationally inexpensive and gives a good biological interpretability.

An approximate loss function for SVM using concepts of logistic regression was proposed by Zhang *et al.* [31]. This function uses a sequence of smooth

functions for iterations to uniformly converge to SVM objective function. The approximate loss function L is given by

$$\mathcal{L}(x, y : w) = \frac{1}{\lambda}\ln\left(1 + \exp\left(-\lambda\left(yw^T x - 1\right)\right)\right) \tag{2}$$

where λ is a tuning parameter and w denotes the weights determining the discriminant. Instead of using a standard 0-1 loss function in SVM, the use of (2) leads to the following penalized objective function:

$$\begin{aligned}\mathcal{L}_P(x, y : w) = &\sum_{j=1}^{m}\frac{1}{\lambda}\ln\left(1 + \exp\left(\lambda\left(1 - y_j w^T x_j\right)\right)\right) \\ &+ \eta\left\|w\right\|^2\end{aligned} \tag{3}$$

where η denotes the sensitivity parameter.

Setting the partial derivation of (3) with respect to each gene i to zero,

$$\begin{aligned}w &= \sum_{j=1}^{m}\frac{1}{2\eta}\frac{\exp\left(\lambda\left(1 - y_j w^T x_j\right)\right)}{1 + \exp\left(\lambda\left(1 - y_j w^T x_j\right)\right)} y_j x_j \\ &= \sum_{j=1}^{m}\alpha_j x_i y_i\end{aligned} \tag{4}$$

Like in SVM, the multiplication factor α_j to $y_j x_j$ incorporates the margin information while computing weights. For example, if margin $y_j w^T x_j$ is greater than one, the multiplication factor becomes zero for large value of λ. In a sense, it rejects the contribution of that particular sample point. Hence, based on this property and considering that $\frac{1}{2\eta}$ is a multiplicative factor, we propose following approximation of support vectors:

$$\alpha_j = \frac{\exp\left(\lambda\left(1 - y_j w^T x_j\right)\right)}{1 + \exp\left(\lambda\left(1 - y_j w^T x_j\right)\right)} \tag{5}$$

With respect to SVM-RFE, the standard 0-1 Loss function gives following SVM weight vector [5, 19]

$$w = \sum_{j=1}^{m}\alpha_j^* y_j x_j \tag{6}$$

Comparing (4),(5) and (6), we can represent the SVM induced weight to a particular sample point α_j^* with α_j.

2.3 T-score with Sample Selection (T-SS)

The margin of a data point is defined as the distance from the data point to the discriminant boundary. The margin of jth data sample is given by the term $y_j w^{\mathrm{T}} x_j$. Zhang *et al.* proposed a gradient descent algorithm to determine the

Algorithm 1. Gene ranking using T-score and sample selection

Begin
Gene set $S = \{i\}_{i=1}^{n}$, data D, and ranked list $R = [\,]$;
set λ; $\epsilon = 0.001$
repeat
  Find the set of samples $M \subset D$ using (5) with $\alpha_j > \epsilon$
  if $|M| < 2$ **then**
    $M = D$
  end if
  Compute the ranking r_i using samples in M
  Select the gene $i^* = \arg\min\{r_i\}$
  Update $R = [R; i^*]\,; S = S \setminus \{i^*\}$
until all genes are ranked
end : output R

margin of separation [31]. In order to simplify the computations, we propose to use T-score of each gene as the selection criteria and thereby remove the optimization step in (3). With this idea, we propose an algorithm for simultaneous sample and gene selection, which is described in Algorithm 1.

Sample points are selected using (5) with a small threshold ϵ. Let M_ℓ denote the set of selected sample points in class ℓ. With a given λ value, the samples are selected using the margin information based on T-score. Using only the selected samples, genes are ranked with T-score. A gene with the least absolute score is then removed from the gene set and the whole process is iterated again until all genes are ranked. In other words, the proposed method selects genes in backward elimination manner while selecting the relevant samples. The T-score with sample selection method fails whenever there is less than two relevant data points. In such cases, we revert to all the sample points and compute the ranking scores in that iteration using all training samples.

The margin is determined by using the T-score of individual gene. It has a direct relation with log-odds ratio if the data is normally distributed, which is given by

$$\log \frac{P\,(+|x)}{P\,(-|x)} = x^T \Sigma^{-1} \left(\mu^+ - \mu^-\right) + w_0 \tag{7}$$

Here, w_0 is a bias term and was computed using

$$w_0 = log \frac{\pi^+}{\pi^-} - \frac{1}{2}\mu^{+T} \Sigma^{-1} \mu^+ + \frac{1}{2}\mu^{-T} \Sigma^{-1} \mu^- \tag{8}$$

where π^+ and π^- represent the prior probabilities of respective classes; the Σ represents the covariance matrix. As we assume independence among genes, in (7) and (8), the covariance matrix becomes $\Sigma = \sigma^2 \mathrm{I}$, where I is the identity matrix. In computing a sample margin $yw^{\mathrm{T}}x$, the bias term is included in w. The weights w are normalized before computing the margin of a sample:

$$w = \frac{w}{\|w\|} \tag{9}$$

Table 1. Details on Benchmark Gene Expression Datasets

Dataset	No. of Genes	No. of + Samples	No. of - Samples
Colon	2000	22	40
Leukemia	7129	38	34
Breast	7129	25	47

3 Experiments and Results

3.1 Datasets and Preprocessing

To evaluate the performance of the proposed method, we performed extensive experiments on three benchmark microarray gene expression datasets, namely, colon [32], leukemia [33], and breast cancer [34]. The details of these widely used datasets for evaluating gene ranking methods are given in Table 1.

All the training datasets were normalized to zero mean and unit variance based on gene expressions of a particular gene to implement T-score, SVM-RFE, SVt-RFE, and T-SS. The datasets were normalized using the parameters from the corresponding training dataset only.

3.2 Parameter Estimation

The parameter λ was determined from a set of $\{1, 3, 5, 7, 10\}$ and selected for the best classification accuracy with the selected genes from Algorithm 1. For algorithms like SVt-RFE and SVM-RFE, the selection of training data points depends on the sensitivity η of the linear SVM, which was determined from the finite set $\left\{2^{-20}, 2^{-19}, \ldots, 2^{15}\right\}$, giving the maximum Matthew's correlation coefficient (MCC[1]) on 10-fold cross-validation.

In recursive elimination, we gradually removed genes in each of the iteration. To increase the speed of the numerical simulations with SVt-RFE, SVM-RFE, and T-SS, the following step-wise strategy was employed:

$$\text{No. of genes removed} = \begin{cases} 100 \text{ if } n' \geq 10000 \\ 10 \quad \text{if } 1000 \leq n' < 10000 \\ 1 \quad n' < 1000 \end{cases} \tag{10}$$

where n' is the number of genes in the gene set.

3.3 Performance Evaluation

With five-fold external cross-validation for 20 times, we obtained $B = 100$ sets of gene ranking lists for each dataset. The gene ranking was obtained using T-score, SVM-RFE, SVt-RFE, and T-SS. The test validation was performed using

[1] $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$.

corresponding test set of a gene ranking list. We tested gene subsets starting from the top ranked genes and then successively adding one gene at a time in test subset till the total number of genes in subset equals 100. The performance measures such as accuracy, sensitivity, and specificity were averaged over those 100 trials. The cardinality of the gene subset giving the minimum average test error was reported as the number of genes corresponding to the best classification performance. We also performed pair-wise one-sided t-test to determine if the performance of the T-SS is significantly better over the other methods.

3.4 Stability Analysis

In this section, a similarity based approach is taken to compute the stability of gene selection, which is measured by the average over all pair-wise similarity comparisons among all the ranked gene lists obtained by the method over different subsets of training samples [35].

Let $\left\{D^b\right\}_{b=1}^{B}$ be a set of B sub-samplings of the dataset of the same size and R^b be the b-th rank list of genes. The stability $\mathcal{S}_{\mathcal{D}}$ of the method over the dataset D is given by

$$\mathcal{S}_D = \frac{2}{B(B-1)} \sum_{b=1}^{B} \sum_{b'=b+1}^{B} \mathcal{S}(R^b, R^{b'}) \tag{11}$$

where $\mathcal{S}(R^b, R^{b'})$ is a similarity measure between the gene rank lists R^b and $R^{b'}$ for top n^* genes in both lists. One of the popular measure to find similarities between two gene lists is a Kuncheva index [35] given by

$$\mathcal{S}(R^b, R^{b'}) = \frac{|R^b \bigcap R^{b'}| - {n^*}^2/n}{n^* - {n^*}^2/n} \tag{12}$$

where n denotes total number of genes in a dataset and n^* is the set of the top genes. Kuncheva index has a range between $[-1, 1]$ with large value indicating large number of common genes between the subsets. A negative Kuncheva index denotes an overlap between two subsets by chance. The term $(n^*)^2/n$ corrects for a bias due to chance of selecting common features between two randomly chosen subsets.

3.5 Redundancy Analysis

Apart from stability and performance in classification, we further evaluate gene selection methods for their ability to select non-redundant genes. We use the absolute value of Pearson's correlation coefficient to estimate the redundancy among top-ranked genes in a given dataset. In a gene rank list R^b, we first measure a pair-wise correlation coefficient of top n^* genes, resulting in a $n^* \times n^*$ correlation matrix with each element representing pair-wise similarity. Using the upper triangular matrix, we obtained average of absolute pair-wise correlations, which represents redundancy among those n^* top-ranked genes in rank list R^b.

Table 2. Performance of T-score, SVM-RFE, SV*t*-RFE, and T-SS on Benchmark *Cancer* Datasets

Dataset	Method	T-score	SVM-RFE	SV*t*-RFE	T-SS
Colon	# Genes	83	97	**32**	91
	Accuracy	86.53 ± 9.00	83.47 ± 9.37	86.08 ± 9.44	$\mathbf{87.12 \pm 9.59}$
	Significance	...	$p < 0.001$	...	
	Sensitivity	80.10 ± 19.08	74.35 ± 18.97	78.25 ± 17.84	$\mathbf{80.90 \pm 18.44}$
	Significance	...	$p < 0.001$	$p < 0.05$	
	Specificity	90.00 ± 10.05	88.50 ± 11.47	90.37 ± 10.02	$\mathbf{90.50 \pm 10.22}$
	Significance	...	$p < 0.001$	...	
Leukemia	# Genes	65	**43**	63	49
	Accuracy	96.36 ± 4.72	96.65 ± 4.14	$\mathbf{97.01 \pm 4.09}$	97.00 ± 4.10
	Significance	$p < 0.05$	...	...	
	Sensitivity	94.40 ± 11.04	95.00 ± 9.16	95.20 ± 9.04	$\mathbf{95.40 \pm 8.46}$
	Significance	...	...	...	
	Specificity	97.43 ± 4.83	97.54 ± 4.52	$\mathbf{97.99 \pm 4.18}$	97.88 ± 4.50
	Significance	...	...	...	
Breast	# Genes	**57**	97	87	99
	Accuracy	86.17 ± 11.78	87.67 ± 11.02	87.97 ± 11.35	$\mathbf{89.30 \pm 10.77}$
	Significance	$p < 0.001$	$p < 0.01$	$p < 0.05$	
	Sensitivity	88.80 ± 13.13	$\mathbf{91.20 \pm 13.43}$	89.80 ± 13.48	89.20 ± 12.85
	Significance	...	...	...	
	Specificity	83.45 ± 19.07	84.15 ± 17.99	86.05 ± 18.82	$\mathbf{89.45 \pm 15.42}$
	Significance	$p < 0.001$	$p < 0.001$	$p < 0.01$	

This value is averaged over the total number of gene rankings, i.e., number of bootstrapped trials (B).

Mathematically, the average redundancy among top n^* genes over B trials can be given by,

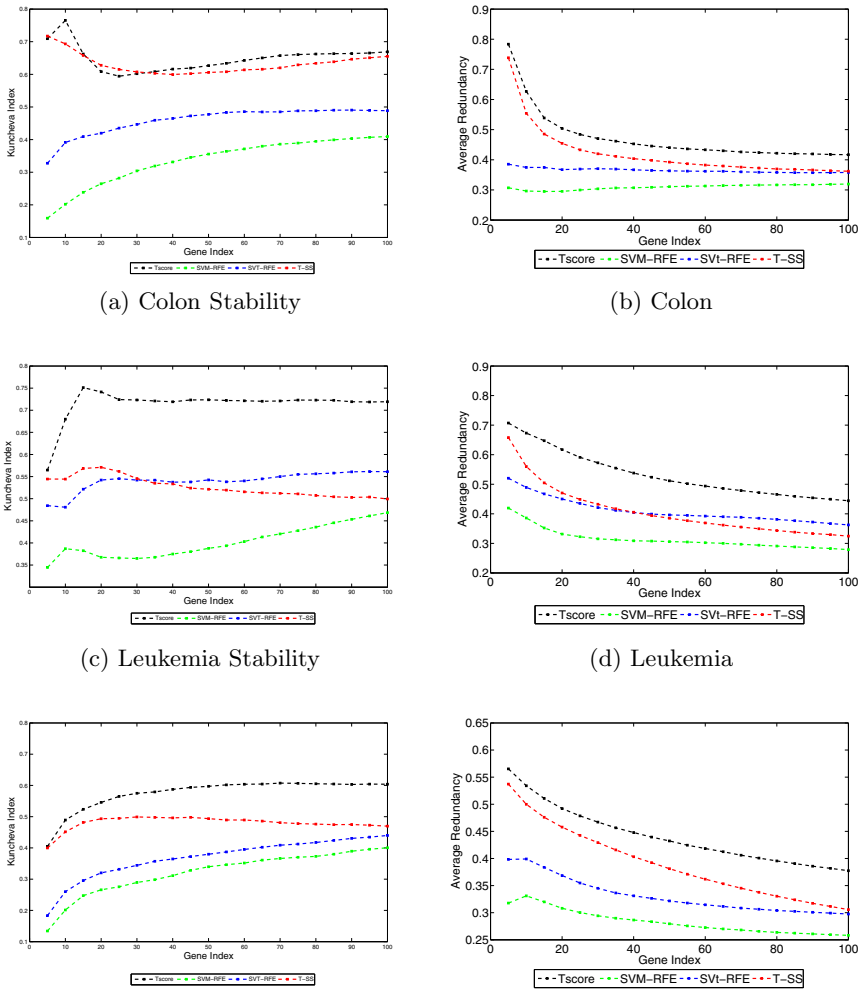
$$\overline{Q} = \frac{1}{B} \sum_{b=1}^{B} \frac{2}{n^* (n^* - 1)} \sum_{i=1}^{n^*-1} \sum_{i'=i+1}^{n^*} |\rho (x_i, x_{i'})| \tag{13}$$

where $|\rho (x_i, x_{i'})|$ is absolute value of Pearson's correlation coefficient between expression values of gene i and i'. In a given dataset, the redundancy analysis is performed over top 100 genes, obtained from various ranking methods.

3.6 Results

A comparison of classification performances of T-score, SVM-RFE, SV*t*-RFE, and T-SS is shown in Table 2. The p-values shown gives the statistical significance of superior performance of T-SS over the other methods. The stability and redundancy plots are depicted in Figure 1.

As seen, the performance of the proposed method is significantly better than the gene ranking by T-score and SVM-RFE methods in at least two datasets. For



(a) Colon Stability (b) Colon

(c) Leukemia Stability (d) Leukemia

(e) Breast Stability (f) Breast

Fig. 1. Plots of Stability and Redundacy against the number of selected genes on benchmark expression datasets

breast cancer dataset, the proposed algorithm outperforms all the other methods. Importantly, the stability plots shows that the proposed method is more stable than SVM-RFE and SVt-RFE methods for top-ranked genes. Comparing redundancy, T-score gave highly redundant top-ranked genes while SVM-RFE returned the least redundant genes. Genes from T-SS and SVt-RFE methods have intermediate redundancy. The numbers of genes selected by T-SS were higher in most cases.

4 Discussion and Conclusion

This paper proposed a sample selection criterion using a modified logistic regression loss function and a backward elimination based gene ranking algorithm. The method selects sample points iteratively before ranking genes using the T-score in each iteration. The performance was evaluated on a number of benchmark datasets and results showed not only promise in the classification results but also the superior stability of the method.

For selection of samples, the approaches involving standard SVM, such as SVt-RFE, are computationally expensive and involves computational complexity of the order of $O(\max(n, m)m^2)$ [36]. If a single gene is removed in each iteration, we need to train SVM for n number of times. On the other hand, selecting samples on the margin with T-score by using an approximation to SVM lost function, the speed is improved. A standard T-score have computational complexity of order of $O(nm)$ [36], so our proposed algorithm is approximate to this complexity compared to SVM-RFE and SVt-RFE.

In two-class classification, as the standard T-score ranks genes independently, it has the highest stability and no penalization for redundancy in gene selection among the other methods tested in the experiments. As SVM-RFE does not treat genes independently and penalizes for redundant genes [37], it is less stable and robust to the variations of training samples. The proposed method not only performs better in classification but retains independence among genes while ranking. This leads to better stability compared to SVM based approaches. Following [37], the proposed sample selection criterion may induce some penalization for the redundant genes. This is evident with reduced redundancy compared to T-score method.

In conclusion, the proposed method is a simple yet efficient criterion for sample selection. Simultaneous sample and gene selection algorithms significantly outperform both T-score and SVM-RFE methods on at least two benchmark datasets. Along with better classification, the proposed method was computationally efficient and highly stable. This suggests that sample selection indeed plays an important role in gene selection. As future of this work, one may want to penalize for redundancy among genes in the cost function as it would improve the stability and performance of tissue classification.

Acknowledgments. This work is supported by a AcRF Tier 2 grant MOE2010-T2-1-056 (ARC 9/10), Ministry of Education, Singapore.

References

1. Inza, I., Larranaga, P., Blanco, R., Cerrolaza, A.: Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence Medicine 31, 91–103 (2004)
2. Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., Nowe, A.: A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 9(4), 1106–1119 (2012)

3. Mundra, P.A., Rajapakse, J.C.: Svm-rfe with mrmr filter for gene selection. IEEE Transactions on Nanobioscience 9(1), 31–37 (2010)
4. Rajapakse, J.C., Mundra, P.A.: Multiclass gene selection using pareto-fronts. IEEE/ACM Transactions on Computational Biology and Bioinformatics (accepted, 2013)
5. Guyon, I., Weston, J., Barhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
6. Cavill, R., Keun, H., Holmes, E., Lindon, J., Nicholson, J., Ebbels, T.: Genetic algorithms for simultaneous variable and sample selection in metabonomics. Bioinformatics 25(1), 112–118 (2009)
7. Chakraborty, S.: Simultaneous cancer classification and gene selection with bayesian nearest neighbor method: An integrated approach. Computational Statistics & Data Analysis 53(4), 1462–1474 (2009)
8. Hapfelmeier, A., Ulm, K.: A new variable selection approach using random forests. Computational Statistics & Data Analysis 60, 50–69 (2013)
9. Kira, K., Rendell, L.A.: A feature selection problem: traditional methods and a new algorithm. In: Proc. of the 10th National Conference on Artificial Intelligence, pp. 129–134 (1992)
10. Wang, Y., Tetko, I., Hall, M., Frank, E., Facius, A., Mayer, K., Mewes, H.: Gene selection from microarray data for cancer classification - a machine learning approach. Computational Biology and Chemistry 29, 37–46 (2005)
11. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J Bioinformatics Computational Biology 3, 185–205 (2005)
12. Tang, Y., Zhang, Y.Q., Huang, Z.: Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. IEEE Trans on Computational Biology and Bioinformatics 4(3), 365–381 (2007)
13. Tang, Y., Zhang, Y.Q., Huang, Z., Hu, X., Zhao, Y.: Recursive fuzzy granulation for gene subset extraction and cancer classification. IEEE Trans on Information Technology in Biomedicine 12(6), 723–730 (2008)
14. Kai-Bo, D., Rajapakse, J., Wang, H., Azuaje, F.: Multiple SVM-RFE for gene selection in cancer classification with expression data. IEEE Trans Nanobioscience 4, 228–234 (2005)
15. Yoon, S., Kim, S.: Adaboost-based multiple svm-rfe for classification of mammograms in ddsm. BMC Medical Informatics and Decision Making 9(S1), 693–708 (2009)
16. Abeel, T., Helleputte, T., Van de Peer, Y., Sayes, Y., et al.: Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 26(3), 392–398 (2010)
17. Diaz-Uriarte, R., Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7, 3 (2006)
18. Zou, H., Hastie, T.: The regularization and variable selection via the elastic net. J. Royal Stat. Society B 67, 301–320 (2005)
19. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience Publications (1998)
20. Freund, Y., Schapire, R.: A short introduction to boosting. J. Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
21. Clarke, R., Ressom, H., Wang, A., Xuan, J., et al.: The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nature Reviews Cancer 8, 37–49 (2008)
22. Han, Y., Yu, L.: A variance reduction framework for stable feature selection. In: Proc. of the 10th IEEE International Conference on Data Mining (2010)

23. Liu, H., Motoda, H., Yu, L.: A selective sampling approach to active feature selection. Artificial Intelligence 159, 49–74 (2004)
24. Pechenizkiy, M., Puuronen, S., Tsymbal, A.: The impact of sample reduction on PCA-based feature extraction for supervised learning. In: Proc. of the 21st ACM Symposium on Applied Computing, pp. 553–558 (2006)
25. Shen, Q., Mei, Z., Ye, B.X.: Simultaneous genes and training samples selection by modified particle swarm optimization for gene expression data classification. Computers in Biology and Medicine 39, 646–649 (2009)
26. Lei, Y., Yue, H., Berens, M.: Stable gene selection from microarray data via sample weighting. IEEE Transactions on Computational Biology and Bioinformatics 9(1), 262–272 (2012)
27. Somol, P., Novovicova, J.: Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. IEEE Transactions on Pattern Analysis and machine intelligence 32(11), 1921–1939 (2010)
28. Haury, A.C., Gestraud, P., Vert, J.P.: The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. Plos One 6(12), e28210 (2011)
29. Mundra, P.A., Rajapakse, J.C.: Gene and sample selection for cancer classification with support vectors based t-statistic. Neurocomputing 73(13-15), 2353–2362 (2010)
30. Mundra, P.A., Rajapakse, J.C.: Support vector based T-score for gene ranking. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) PRIB 2008. LNCS (LNBI), vol. 5265, pp. 144–153. Springer, Heidelberg (2008)
31. Zhang, J., Jin, R., Yang, Y., Hauptmann, A.: Modified logistic regressionl an approximation to svm and its applications in large-scale text categorization. In: Proceedings of 20th International Conference on Machine Learning, ICML 2003 (2003)
32. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS 96, 6745–6750 (1999)
33. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: Class discovery and class prediction by gene expression. Science 286, 531–537 (1999)
34. West, M., Blanchette, C., Dressman, H., et al.: Predicting the clinical status of human breast cancer by using gene expression profiles. Proceedings of National Academy of sciences 98(20), 11462–11467 (2001)
35. Kuncheva, L.: A stability index for feature selection. In: Proceedings of the 25th IASTED International Conference on Artificial Intelligence and Applications, pp. 390–395 (2007)
36. Guyon, I., Elisseeff, A.: An introduction to feature extraction. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) Feature Extraction, Foundations and Applications. STUDFUZZ, pp. 1–25. Springer, Heidelberg (2006)
37. Li, F., Yang, Y.: Analysis of recursive gene selection approaches from microarray data. Bioinformatics 21(19), 3741–3747 (2005)

Versatile Sparse Matrix Factorization and Its Applications in High-Dimensional Biological Data Analysis

Yifeng Li and Alioune Ngom

School of Computer Science, University of Windsor,
401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{li11112c,angom}@uwindsor.ca

Abstract. Non-negative matrix factorization and sparse representation models have been successfully applied in high-throughput biological data analysis. In this paper, we propose our *versatile sparse matrix factorization* (VSMF) model for biological data mining. We show that many well-known sparse models are specific cases of VSMF. Through tuning parameters, sparsity, smoothness, and non-negativity can be easily controlled in VSMF. Our computational experiments corroborate the advantages of VSMF.

Keywords: versatile sparse matrix factorization, non-negative matrix factorization, sparse representation,feature extraction, feature selection, biological processes identification.

1 Introduction

Non-negative matrix factorization (NMF) [10] and the wider concept – *sparse representation* (SR) [6] are sparse matrix factorization models that decompose a matrix into a basis matrix and coefficient matrix. They have been applied in many fields of bioinformatics including clustering [3] and biclustering [4], sample prediction [15], biological process identification [9], and transcriptional regulatory network inference [16]. Many variants of NMF and SR have been invented for various situations. Semi-NMF is proposed in [5] for data of mixed signs. Sparse NMF is introduced to guarantee sparse results [8]. We propose kernel NMF in [13] to deal with nonlinearity in microarray data. Kernel NMF also works for relational data. Negative values are allowed in the coefficient matrix of l_1-regularized sparse representation (l_1-SR) models [15]. However, the following challenges have not been well addressed. First, an unified model is very necessary for these variants from both theoretical and practical perspectives. Second, sparsity is usually constrained on the coefficient matrix and the sparsity of basis matrix is not guaranteed in most sparse models. Third, l_1-norm is the most popular way to induce sparsity. However, it does not guarantee that a group of correlated variables can be selected or discarded simultaneously.

In this paper, in order to address these challenges, we propose our *versatile sparse matrix factorization* (VSMF) model. The contributions of this study includes

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 91–101, 2013.

1. With its six parameters, VSMF can easily control sparsity, smoothness, and non-negativity on both basis matrix and coefficient matrix. VSMF is thus a generic model. The standard NMF, semi-NMF, sparse NMF, kernel NMF and l_1-SR models are specific cases of VSMF.
2. We devise multiplicative update rules and active-set algorithms for the optimization of VSMF. Analytical solutions, which are useful for kernelization, are also discussed.
3. We demonstrate the usefulness of VSMF in bioinformatics.

The rest of this paper is organized as follows: We first summarize the variants of NMF or SR models in Section 2. Next, we present our VSMF model and its optimization in Section 3. After that, several biological applications of VSMF are demonstrated in Section 4. Finally, we draw our conclusions and mention future works.

2 Related Work

Hereafter, we use the following notations. The training set is denoted by $[\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n] = \boldsymbol{D} \in \mathbb{R}^{m \times n}$, where m and n are the numbers of features and samples respectively. The basis matrix is represented as $[\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k] = \boldsymbol{A} \in \mathbb{R}^{m \times k}$, where $k < \min\{m, n\}$ is the number of basis vectors (or factors). The coefficient matrix is denoted by $[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n] = \boldsymbol{Y} \in \mathbb{R}^{k \times n}$. Given $\boldsymbol{D}$, the task of sparse matrix factorization is to find $\boldsymbol{A}$ and $\boldsymbol{Y}$ such that $\boldsymbol{D} \approx \boldsymbol{AY}$, where at least one factors among $\boldsymbol{A}$ and $\boldsymbol{Y}$ should be sparse.

For the convenience of discussion, we summarize the existing sparse matrix factorization models in Table 1. It is impossible to enumerate all existing works in this direction, therefore all models mentioned in this table are the most representative ones. The training data $\boldsymbol{D}$ must be non-negative for the standard NMF and sparse NMF. For sparse NMF, α and λ are two non-negative parameters. For kernel NMF and l_1-SR, $\phi(\cdot)$ is a function that maps the training samples into a high-dimensional feature space. $\phi(\boldsymbol{D})$ is the training samples in this feature space. $\boldsymbol{A}_\phi$ is the basis matrix in this feature space.

Table 1. The Existing NMF and SR Models

NMF/SR	Equations
Standard NMF [10]	$\min_{\boldsymbol{A},\boldsymbol{Y}} \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{AY}\|_F^2$ s.t. $\boldsymbol{A}, \boldsymbol{Y} \geq 0$
Semi-NMF [5]	$\min_{\boldsymbol{A},\boldsymbol{Y}} \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{AY}\|_F^2$ s.t. $\boldsymbol{Y} \geq 0$
Sparse NMF [8]	$\min_{\boldsymbol{A},\boldsymbol{Y}} \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{AY}\|_F^2 + \frac{\alpha}{2}\sum_{i=1}^{k}\|\boldsymbol{a}_i\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\|\boldsymbol{y}_i\|_1^2$ s.t. $\boldsymbol{A}, \boldsymbol{Y} \geq 0$
Kernel NMF [13, 15]	$\min_{\boldsymbol{A}_\phi,\boldsymbol{Y}} \frac{1}{2}\|\phi(\boldsymbol{D}) - \boldsymbol{A}_\phi\boldsymbol{Y}\|_F^2 + \frac{\alpha}{2}\sum_{i=1}^{k}\|\phi(\boldsymbol{a}_i)\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\|\boldsymbol{y}_i\|_1$ s.t. $\boldsymbol{Y} \geq 0$
l_1-SR [15]	$\min_{\boldsymbol{A}_\phi,\boldsymbol{Y}} \frac{1}{2}\|\phi(\boldsymbol{D}) - \boldsymbol{A}_\phi\boldsymbol{Y}\|_F^2 + \frac{\alpha}{2}\sum_{i=1}^{k}\|\phi(\boldsymbol{a}_i)\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\|\boldsymbol{y}_i\|_1$

3 Method

In this section, we first present our versatile sparse matrix factorization model. Then, we give optimization algorithms for this model.

3.1 The Versatile Sparse Matrix Factorization Model

Our *versatile sparse matrix factorization* (VSMF) model can be expressed in the following equation:

$$\begin{aligned}\min_{\boldsymbol{A},\boldsymbol{Y}} f(\boldsymbol{A},\boldsymbol{Y}) &= \frac{1}{2}\|\boldsymbol{D}-\boldsymbol{A}\boldsymbol{Y}\|_F^2 + \sum_{i=1}^{k}\left(\frac{\alpha_2}{2}\|\boldsymbol{a}_i\|_2^2 + \alpha_1\|\boldsymbol{a}_i\|_1\right) \\ &+ \sum_{i=1}^{n}\left(\frac{\lambda_2}{2}\|\boldsymbol{y}_i\|_2^2 + \lambda_1\|\boldsymbol{y}_i\|_1\right) \\ \text{s.t.} &\begin{cases}\text{if } t_1 = 1 & \boldsymbol{A} \geq 0 \\ \text{if } t_2 = 1 & \boldsymbol{Y} \geq 0\end{cases},\end{aligned} \tag{1}$$

where parameter $\alpha_1 \geq 0$ controls the sparsity of the basis vectors, parameter $\alpha_2 \geq 0$ controls the smoothness and scale of the basis vectors, parameter $\lambda_1 \geq 0$ controls the sparsity of the coefficient vectors, parameter $\lambda_2 \geq 0$ controls the smoothness of the coefficient vectors, parameters t_1 and t_2 are boolean variables (0: false, 1: true) that indicate if non-negativity should be enforced on $\boldsymbol{A}$ and $\boldsymbol{Y}$, respectively.

One advantage of VSMF is that both l_1 and l_2-norms can be used on both basis matrix and coefficient matrix. In VSMF, l_1-norms are used to induce sparse basis vectors and coefficient vectors. However, the drawback of l_1-norm is that correlated variables may not be simultaneously non-zero in the induced sparse result. This is because l_1-norm is able to produce sparse but non-smooth result. It is known that l_2-norm is able to obtain smooth but not sparse result. Combining both norms has been proven that correlated variables can be selected or removed simultaneously [18]. In addition to the smoothness of l_2-norm, another benefit of l_2-norm is that the scale of each vector can be restricted. This can avoid the scale interchange between the basis matrix and coefficient matrix. Another advantage of VSMF is that the non-negativity constraint can be switched off/on for either basis matrix or coefficient matrix. If the training data are non-negative, it is usually necessary that the basis matrix should be non-negative as well. In some situations, non-negativity is also needed on the coefficient matrix for better performance and better interpretation.

It can be easily seen that the standard NMF, semi-NMF, and sparse-NMFs are special cases of VSMF. If $\alpha_1 = \alpha_2 = \lambda_1 = \lambda_2 = 0$ and $t_1 = t_2 = 1$, VSMF is reduced to the standard NMF proposed in [10]. If $\alpha_1 = \alpha_2 = \lambda_1 = \lambda_2 = 0$ and $t_1 = 0$ and $t_2 = 1$, then VSMF becomes semi-NMF proposed in [5]. If $\alpha_1 = \lambda_2 = 0$, $\alpha_2, \lambda_1 \neq 0$, and $t_1 = t_2 = 1$, then VSMF is equivalent to the sparse-NMF proposed in [8]. When α_1 is set to zero, VSMF can be kernelized [15].

Sparse matrix factorization is a low-rank approximation problem. The number of ranks, that is k, is crucial for good performance of an analysis. Selecting k is still an open problem in both statistical inference and machine learning. We propose an adaptive rank selection method for VSMF. We base our idea on the sparsity of columns of $\boldsymbol{A}$ and $\boldsymbol{Y}$. We first set k to a relatively large integer. During the optimization of VSMF, if a column of $\boldsymbol{A}$ or a row of $\boldsymbol{Y}$ is null due to the sparsity controlled by the corresponding parameters, then both of the column of $\boldsymbol{A}$ and the row of $\boldsymbol{Y}$ corresponding to this null

factor are removed. Therefore k is reduced. When the optimization terminates, we can obtain the correct k corresponding to the current sparsity controlling parameters.

3.2 Optimization

Like most of NMF and SR models, the optimization of VSMF is non-convex. The most popular scheme to optimize these models are the block-coordinate descent method [2]. The basic idea of this scheme is in the following. $\boldsymbol{A}$ and $\boldsymbol{Y}$ are updated iteratively and alternatingly. In each iteration, $\boldsymbol{A}$ is updated while keeping $\boldsymbol{Y}$ fixed; then $\boldsymbol{A}$ is fixed and $\boldsymbol{Y}$ is updated. Based on this scheme, we devise the multiplicative update rules and active-set algorithms for VSMF. These two algorithms are given below.

Multiplicative Update Rules. If both $\boldsymbol{A}$ and $\boldsymbol{Y}$ are non-negative, we can equivalently rewrite $f(\boldsymbol{A}, \boldsymbol{Y})$ in Equation (1) to

$$\frac{1}{2}\|\boldsymbol{D} - \boldsymbol{A}\boldsymbol{Y}\|_F^2 + \frac{\alpha_2}{2}\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) + \alpha_1\mathrm{tr}(\boldsymbol{E}_1^{\mathrm{T}}\boldsymbol{A}) + \frac{\lambda_2}{2}\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) + \lambda_1\mathrm{tr}(\boldsymbol{E}_2^{\mathrm{T}}\boldsymbol{Y}), \quad (2)$$

where $\boldsymbol{E}_1 \in \{1\}^{m\times k}$, and $\boldsymbol{E}_2 \in \{1\}^{k\times n}$. Fixing $\boldsymbol{A}$, updating $\boldsymbol{Y}$ can hence be expressed as

$$\begin{aligned} &\min_{\boldsymbol{Y}} f(\boldsymbol{Y}) = \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{A}\boldsymbol{Y}\|_F^2 + \frac{\lambda_2}{2}\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) + \lambda_1\mathrm{tr}(\boldsymbol{E}_2^{\mathrm{T}}\boldsymbol{Y}) \\ &\text{s.t. } \boldsymbol{Y} \geq 0. \end{aligned} \quad (3)$$

Similarly, Fixing $\boldsymbol{Y}$, updating $\boldsymbol{A}$ can be expressed as

$$\begin{aligned} &\min_{\boldsymbol{A}} f(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{A}\boldsymbol{Y}\|_F^2 + \frac{\alpha_2}{2}\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) + \alpha_1\mathrm{tr}(\boldsymbol{E}_1^{\mathrm{T}}\boldsymbol{A}) \\ &\text{s.t. } \boldsymbol{A} \geq 0. \end{aligned} \quad (4)$$

We design the following multiplicative update rules for VSMF model in the case of $t_1 = t_2 = 1$:

$$\begin{cases} \boldsymbol{A} = \boldsymbol{A} * \frac{\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}}{\boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{A} + \alpha_1} \\ \boldsymbol{Y} = \boldsymbol{Y} * \frac{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}}{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \lambda_2\boldsymbol{Y} + \lambda_1} \end{cases}, \quad (5)$$

where $\boldsymbol{A} * \boldsymbol{B}$ and $\frac{\boldsymbol{A}}{\boldsymbol{B}}$ are element-wise multiplication and division between matrix $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. This algorithm is a gradient-descent based method. Both rules are derived in the following.

For Equation (3), the first-order update rule of $\boldsymbol{Y}$ should be generally

$$\boldsymbol{Y} = \boldsymbol{Y} - \boldsymbol{\eta}_2 * \frac{\partial f(\boldsymbol{Y})}{\partial \boldsymbol{Y}}. \quad (6)$$

where matrix $\boldsymbol{\eta}_2$ is step. We take the derivative of $f(\boldsymbol{Y})$, in Equation (3), with respect to $\boldsymbol{Y}$:

$$\frac{\partial f(\boldsymbol{Y})}{\partial \boldsymbol{Y}} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{D} + \lambda_2\boldsymbol{Y} + \lambda_1\boldsymbol{E}_2. \quad (7)$$

And we let the step $\boldsymbol{\eta}_2$ to be

$$\boldsymbol{\eta}_2 = \frac{\boldsymbol{Y}}{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \lambda_2\boldsymbol{Y} + \lambda_1\boldsymbol{E}_2}. \tag{8}$$

Substituting Equations (7) and (8) into Equation (6), we have

$$\boldsymbol{Y} = \boldsymbol{Y} * \frac{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}}{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \lambda_2\boldsymbol{Y} + \lambda_1\boldsymbol{E}_2}. \tag{9}$$

Similarly, for Equation (4), the first-order update rule of $\boldsymbol{A}$ should be generally

$$\boldsymbol{A} = \boldsymbol{A} - \boldsymbol{\eta}_1 * \frac{\partial f(\boldsymbol{A})}{\partial \boldsymbol{A}}. \tag{10}$$

We take the derivative of $f(\boldsymbol{A})$, in Equation (4), with respect to $\boldsymbol{A}$:

$$\frac{\partial f(\boldsymbol{A})}{\partial \boldsymbol{A}} = \boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{Y} + \alpha_2\boldsymbol{A} + \alpha_1\boldsymbol{E}_1. \tag{11}$$

And we let the step to be

$$\boldsymbol{\eta}_1 = \frac{\boldsymbol{A}}{\boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{A} + \alpha_1\boldsymbol{E}_1}. \tag{12}$$

Substituting Equations (11) and (12) into Equation (10), we have

$$\boldsymbol{A} = \boldsymbol{A} * \frac{\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}}{\boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{A} + \alpha_1}. \tag{13}$$

If we let $\alpha_1 = \alpha_2 = \lambda_1 = \lambda_2 = 0$, then the update rules in Equations (5) becomes the update rules of the standard NMF [11]. We can find that enforcing sparsity and smoothness on both basis matrix and coefficient matrix does not increase the time-complexity.

Active-Set Quadratic Programming. The multiplicative update rules above only works under the condition that both $\boldsymbol{A}$ and $\boldsymbol{Y}$ are non-negative. We devise active-set algorithms which allow us to relax the non-negativity constraints. We now show that when $t_1($ or $t_2) = 1$, $\boldsymbol{A}$ (or $\boldsymbol{Y}$) can be updated by our active-set *non-negative quadratic programming* (NNQP) algorithm; when $t_1($ or $t_2) = 0$, $\boldsymbol{A}$ (or $\boldsymbol{Y}$) can be updated by our active-set l_1*-regularized QP* (1_1QP) algorithm.

If $t_2 = 1$, the objective in Equation (3) can be rewritten as:

$$\begin{aligned} f(\boldsymbol{Y}) &= \mathrm{tr}(\frac{1}{2}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \frac{\lambda_2}{2}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} + \lambda_1\boldsymbol{E}_2^{\mathrm{T}}\boldsymbol{Y}) \\ &= \mathrm{tr}(\frac{1}{2}\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_2\boldsymbol{I})\boldsymbol{Y} + (\lambda_1\boldsymbol{E}_2^{\mathrm{T}} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{Y} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}) \\ &= \sum_{i=1}^{n} \frac{1}{2}\boldsymbol{y}_i^{\mathrm{T}}\boldsymbol{H}_2\boldsymbol{y}_i + \boldsymbol{g}_{(2)i}^{\mathrm{T}}\boldsymbol{y}_i + \frac{1}{2}\boldsymbol{d}_i^{\mathrm{T}}\boldsymbol{d}_i, \end{aligned} \tag{14}$$

where $\boldsymbol{H}_2 = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_2\boldsymbol{I}$, and $\boldsymbol{g}_{(2)i} = \lambda_1 - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{d}_i$ and $\boldsymbol{G}_{(2)} = \lambda_1 - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}$. Therefore, we can see that updating non-negative $\boldsymbol{Y}$ is multiple NNPQ problem. We proposed a parallel active-set algorithm for NNQP in [15]. This algorithm can be used to solve the problem in Equation (14).

If $t_2 = 0$, the objective of updating $\boldsymbol{Y}$ can be reformulated as:

$$\begin{aligned} f(\boldsymbol{Y}) &= \mathrm{tr}(\frac{1}{2}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} - \boldsymbol{D}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Y} + \frac{\lambda_2}{2}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) + \lambda_1\|\boldsymbol{Y}\|_1 \\ &= \mathrm{tr}(\frac{1}{2}\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_2\boldsymbol{I})\boldsymbol{Y} + (-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{Y} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}) + \lambda_1\|\boldsymbol{Y}\|_1 \\ &= \sum_{i=1}^{n} \frac{1}{2}\boldsymbol{y}_i^{\mathrm{T}}\boldsymbol{H}_2\boldsymbol{y}_i + \boldsymbol{g}_{(2)i}^{\mathrm{T}}\boldsymbol{y}_i + \lambda_1\|\boldsymbol{y}_i\|_1 + \frac{1}{2}\boldsymbol{d}_i^{\mathrm{T}}\boldsymbol{d}_i, \end{aligned} \tag{15}$$

where $\boldsymbol{H}_2 = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_2\boldsymbol{I}$, and $\boldsymbol{g}_{(2)i} = -\boldsymbol{A}^{\mathrm{T}}\boldsymbol{d}_i$ and $\boldsymbol{G}_{(2)} = -\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}$. This is a l_1QP problem which can be solved by our active-set l_1QP algorithm proposed in [15].

Similarly, if $t_1 = 1$, $f(\boldsymbol{A})$ in Equation (3) can be expressed as

$$\begin{aligned} f(\boldsymbol{A}) &= \mathrm{tr}(\frac{1}{2}\boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} - \boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} + \frac{\alpha_2}{2}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} + \alpha_1\boldsymbol{E}_1^{\mathrm{T}}\boldsymbol{A}) \\ &= \mathrm{tr}(\frac{1}{2}\boldsymbol{A}(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}} + (\alpha_1\boldsymbol{E}_1^{\mathrm{T}} - \boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}})\boldsymbol{A}^{\mathrm{T}} + \frac{1}{2}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}) \\ &= \sum_{i=1}^{m} \frac{1}{2}\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{H}_{\boldsymbol{1}}\boldsymbol{w}_i + \boldsymbol{g}_{(1)i}^{\mathrm{T}}\boldsymbol{w}_i + \frac{1}{2}\boldsymbol{D}_{i,:}(\boldsymbol{D}^{\mathrm{T}})_{:,i}, \end{aligned} \tag{16}$$

where $\boldsymbol{W} = \boldsymbol{A}^{\mathrm{T}}$, $\boldsymbol{H}_{\boldsymbol{1}} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{I}$, $\boldsymbol{g}_{(1)i} = \alpha_1 - \boldsymbol{Y}(\boldsymbol{D}^{\mathrm{T}})_{:,i}$ and $\boldsymbol{G}_{(1)} = \alpha_1 - \boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}}$. Again, it can be seen that this problem is also a NNQP problem.

If $t_1 = 0$, the objective of updating $\boldsymbol{A}$ can be written as

$$\begin{aligned} f(\boldsymbol{A}) &= \mathrm{tr}(\frac{1}{2}\boldsymbol{A}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} + \frac{1}{2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} - \boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} + \frac{\alpha_2}{2}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}) + \alpha_1\|\boldsymbol{A}\|_1 \\ &= \mathrm{tr}(\frac{1}{2}\boldsymbol{A}(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{I})\boldsymbol{A}^{\mathrm{T}} + (-\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}})\boldsymbol{A}^{\mathrm{T}} + \frac{1}{2}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}) + \alpha_1\|\boldsymbol{A}^{\mathrm{T}}\|_1 \\ &= \sum_{i=1}^{m} \frac{1}{2}\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{H}_{\boldsymbol{1}}\boldsymbol{w}_i + \boldsymbol{g}_{(1)i}^{\mathrm{T}}\boldsymbol{w}_i + \alpha_1\|\boldsymbol{w}_i\|_1 + \frac{1}{2}\boldsymbol{D}_{i,:}(\boldsymbol{D}^{\mathrm{T}})_{:,i}, \end{aligned} \tag{17}$$

where $\boldsymbol{W} = \boldsymbol{A}^{\mathrm{T}}$, $\boldsymbol{H}_{\boldsymbol{1}} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{I}$, $\boldsymbol{g}_{(1)i} = -\boldsymbol{Y}(\boldsymbol{D}^{\mathrm{T}})_{:,i}$ and $\boldsymbol{G}_{(1)} = -\boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}}$. This is also a l_1QP problem that can be solved by our active-set l_1QP algorithm [15].

Analytical Solutions and Kernelization. If $t_2 = 0$ and $\lambda_1 = 0$, from $\frac{\partial f(\boldsymbol{Y})}{\partial \boldsymbol{Y}} = 0$, $\boldsymbol{Y}$ can be updated analytically:

$$\boldsymbol{Y} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_2\boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D} = \boldsymbol{A}^{\ddagger}\boldsymbol{D}. \tag{18}$$

From the previous section, we can see that only $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ and $\boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}}$ are required to update $\boldsymbol{A}$. According to Equation (18), $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ and $\boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}}$ can be expressed as

$$\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} = \boldsymbol{A}^{\ddagger}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}(\boldsymbol{A}^{\ddagger})^{\mathrm{T}}. \tag{19}$$

$$\boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}} = \boldsymbol{A}^{\ddagger}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}. \tag{20}$$

We can see that in this situation, updating $\boldsymbol{A}$ only requires the previous value of $\boldsymbol{A}$ and the inner products of rows of $\boldsymbol{D}$.

Similarly, if $t_1 = 0$ and $\alpha_1 = 0$, $\boldsymbol{A}$ can be updated analytically:

$$\boldsymbol{A} = \boldsymbol{D}\boldsymbol{Y}^{\ddagger}, \tag{21}$$

where $\boldsymbol{Y}^{\ddagger} = \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} + \alpha_2\boldsymbol{I})^{-1}$. From the previous section, we know that updating $\boldsymbol{Y}$ only requires the inner products $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ and $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}$. They can be updated by the following equations:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = (\boldsymbol{Y}^{\ddagger})^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{Y}^{\ddagger}. \tag{22}$$

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D} = (\boldsymbol{Y}^{\ddagger})^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}. \tag{23}$$

Due to the analytical solution of $\boldsymbol{A}$, updating $\boldsymbol{Y}$ only requires the previous value of $\boldsymbol{Y}$ and the inner products of columns of $\boldsymbol{D}$.

These analytical solutions have two advantages. First, the corresponding matrix can be easily updated without resorting to any numerical solver. Second, we can see that only inner products are needed to update $\boldsymbol{Y}$ (or $\boldsymbol{A}$), when $\boldsymbol{A}$ (or $\boldsymbol{Y}$) can be analytically obtained. Using this property, we can obtain the kernel version of VSMF, which are described in the following. In sparse representation, at least one matrix among $\boldsymbol{A}$ and $\boldsymbol{Y}$ must be sparse. That is the analytical solutions in Equations (18) and (21) can not be used simultaneously. In practice, if each column of the training data $\boldsymbol{D}$ is the object to be mapped in high-dimensional feature space, we can analytically update $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ (or the corresponding kernel version $k(\boldsymbol{A}, \boldsymbol{A}) = (\phi(\boldsymbol{A}))^{\mathrm{T}}\phi(\boldsymbol{A})$ where $k(\cdot, \cdot)$ is a kernel function corresponding to $\phi(\cdot)$) and $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}$ (or $k(\boldsymbol{A}, \boldsymbol{D}) = (\phi(\boldsymbol{A}))^{\mathrm{T}}\phi(\boldsymbol{D})$), and then update $\boldsymbol{Y}$ via a numerical solver described in the previous section. This leads to the kernel sparse representation proposed in [15]. Alternatively, if each row of $\boldsymbol{D}$ is the object to be mapped in high-dimensional feature space, $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ and $\boldsymbol{Y}\boldsymbol{D}^{\mathrm{T}}$ should be updated analytically, then $\boldsymbol{A}$ is updated by a solver given in the previous section. This leads to an alternative kernel sparse representation model.

4 Computational Experiment

Sparse matrix factorization has a wide ranges of applications in biological data analysis. Technically speaking, these applications are based on clustering, biclustering, feature extraction, classification, and feature selection. In this paper, we give three examples to show that promising performance can be obtained by VSMF for feature extraction, feature selection, and biology process identification. For other applications of NMF, please refer to [14].

4.1 Feature Extraction and Classification

NMF is a successful feature extraction method in bioinformatics [12]. Dimension reduction including feature extraction and feature selection is an important step for classification. We compared the performance of our VSMF (for feature extraction) with NMF on a popular microarray gene expression data – Colon [1]. This data set has 2000 genes (features) and 62 samples. There are two classes in this data set. Each sample is normalized to have unit l_2-norm. We employed 4-fold cross-validation to split the whole data into training and test sets. For each split, features were extracted by NMF or VSMF from the training set. The *nearest neighbor* (NN) classifier was used to predict the class labels of the test set. 4-fold cross-validation was repeated for 20 times. We initialized $k = 8$, thus the actual value of k, after calling VSMF, should be less than or equal to 8. *Radial basis function* (RBF) is used in the kernel VSMF. We set the kernel parameter $\sigma = 2^0$. The mean accuracy, *standard deviation* (STD), computing time, and parameter setting are given in Table 2. The standard NMF obtained a mean accuracy of 0.7645, while the linear VSMF yielded 0.7919. The highest accuracy, 0.7944, is obtained by the kernel VSMF. The kernel VSMF only took 1.3346 seconds, which is faster than the linear VSMF and NMF, because the analytical solution of $\boldsymbol{A}$ can be computed for kernel VSMF. We treat this comparison as a demonstration that tuning the parameters of VSMF may obtain better accuracy than NMF. VSMF can be used for many other types of high-throughput data such as copy number profiles and mass spectrometry data.

Table 2. The Classification Performance of VSMF Compared to The Standard NMF. The time is measure by stopwatch timer (the `tic` and `toc` functions in MATLAB) in seconds.

Method	Accuracy (STD)	Time	Parameters
NN	0.7742(0.0260)	0.0137	
NMF+NN	0.7645(0.0344)	4.3310	
Linear VSMF+NN	0.7919(0.0353)	3.1868	$\alpha_2 = 2^{-3}, \lambda_1 = 2^{-6}, t_1 = t_2 = 1$
Kernel VSMF+NN	0.7944(0.0438)	1.3346	$\alpha_2 = 2^{-3}, \lambda_1 = 2^{-6}, t_1 = t_2 = 1, \sigma = 2^0$

4.2 Feature Selection

VSMF can be applied to feature selection. The basic idea is to make the basis vectors sparse, and then select features that vary dramatically among the basis vectors. In our current study of gene selection, we use the following strategy on the sparse basis matrix $\boldsymbol{A}$. For the i-th row (that is the i-th gene), We denote $\boldsymbol{g}_i = \boldsymbol{A}_{i,:}$. If the maximum value in $\boldsymbol{g}_i$ is greater than $\theta = 10^4$ times of the rest values in $\boldsymbol{g}_i$, then we select this gene, otherwise discard it. We tested this VSMF-based feature selection method on a microarray breast tumor data set which have 13582 genes and 158 samples from five classes [7]. The data were normalized so that each gene has mean zero and STD 1. We used the following parameters of VSMF: $\alpha_1 = 2^4$, $\alpha_2 = 2^0$, $\lambda_1 = 0$, $\lambda_2 = 2^0$, $t_1 = 0$, and $t_2 = 1$. The value of k was initialized by 5. The genes selected were validated by classification performance. We employed 20 runs of 4-fold cross-validation. For

each split of training and test sets, genes were selected using the training set. On the dimension-reduced training set, a linear *support vector machine* (SVM) was learned in order to predict the class labels of the corresponding test set. When using all genes to training SVM, we obtained a mean accuracy of 0.8250 with STD 0.0201. When applying the VSMF-based gene selection, we achieved a mean accuracy of 0.8271 with STD 0.0174. We can see that SVM using our gene selection strategy can obtain similar performance with that of using all genes.

4.3 Biological Process Identification

NMF has been applied on either static gene-sample or time-series microarray data to identify potential biological processes [8, 9, 16, 17]. In our experiment, we run our VSMF on the *Gastrointestinal stromal tumor* (GIST) time-series data to show that VSMF can smooth biological processes compared with the result obtained by the standard NMF. This data set was obtained after the treatment of imatinib mesylate. It has 1336 genes and 9 time points. Each gene time-series is normalized to have unit l_2-norm. The smoothness is controlled by parameter α_2. We set the parameters of VSMF to $\alpha_2 = 2^{-2}$, $\lambda_1 = 2^{-8}$, $\alpha_1 = \lambda_2 = 0$, and $t_1 = t_2 = 1$. The number of factors k was set to 3. The basis vectors of NMF and VSMF are shown at the left and right sides of Fig. 1, respectively. We can see that both of them can reconstruct the falling, rising, and transient patterns identified in [16]. The patterns obtained by VSMF are smoother than those of the standard NMF.

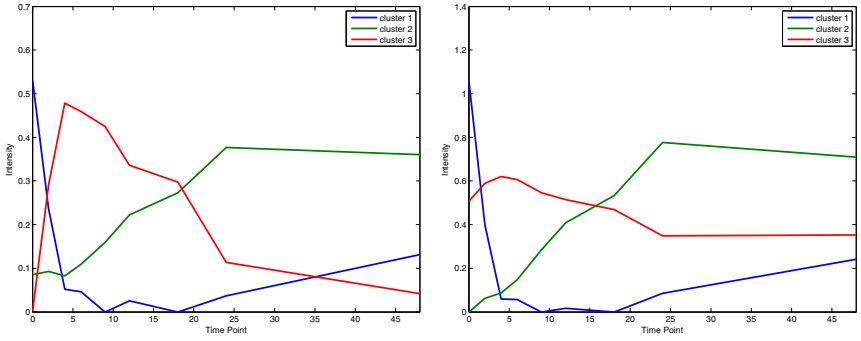


Fig. 1. The Biological processes identified b the standard NMF (*left*) and VSMF (*right*). The result of VSMF is smoother than that of the standard NMF.

5 Conclusions

In this paper, we propose a versatile sparse matrix factorization (VSMF) model for biological data analysis. VSMF is a unified model of many variants of NMF and SR. We give efficient optimization algorithms for VSMF. As shown in our computational demonstrations, many analysis can be conveniently conducted by VSMF for biological

data. The implementation of VSMF can be found in our open-source MATLAB NMF toolbox [14]. The multiplicative-update-rules based VSMF is implemented in function `sparsenmf2rule` and the function `vsmf` includes the implementation of NNQP, l_1QP, and analytical solutions.

We present our on-going work on VSMF in this paper. There remains many interesting challenges in its theoretical and practical aspects. First, there are four key parameters, in the objective of VSMF, which provide flexibility, while rise concerns on the model selection. The two parameters in the constraints can be determined by the signs of a data set. We are working on a guide of parameter selection for VSMF which can be easily tailored for various applications. The value of k is also related with the sparsity, thus we need further investigation on it. Second, increasing the value of α_1 leads to a more sparse basis matrix. This is very helpful for feature selection. We will investigate more effective feature selection method using VSMF. The performance of VSMF for feature selection will be compared statistically with existing approaches. The genes selected will be validated by permutation test and gene set enrichment analysis.

Acknowledgments. This research is supported by IEEE CIS Walter Karplus Summer Research Grant 2010, Ontario Graduate Scholarships 2011-2013, NSERC Grants #RGPIN228117-2011, and several scholarships from The University of Windsor.

References

1. Alon, U.: Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS 96(12), 6745–6750 (1999)
2. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, 2nd edn., Belmont, MA (2008)
3. Brunet, J., Tamayo, P., Golub, T., Mesirov, J.: Metagenes and molecular pattern discovery using matrix factorization. PNAS 101(12), 4164–4169 (2004)
4. Carmona-Saez, P., Pascual-Marqui, R.D., Tirado, F., Carazo, J.M., Pascual-Montano, A.: Biclustering of gene expression data by non-smooth non-negative matrix factorization. BMC Bioinformatics 7, 78 (2006)
5. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. TPAMI 32(1), 45–55 (2010)
6. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, New York (2010)
7. Hu, Z.: The molecular portraits of breast tumors are conserved across microarray platforms. BMC Genomics 7, 96 (2006)
8. Kim, H., Park, H.: Sparse non-negatice matrix factorization via alternating non-negativity-constrained least aquares for microarray data analysis. SIAM J. Matrix Analysis and Applications 23(12), 1495–1502 (2007)
9. Kim, P., Tidor, B.: Subsystem identification through dimensionality reduction of large-scale gene expression data. Genome Research 13, 1706–1718 (2003)
10. Lee, D.D., Seung, S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
11. Lee, D., Seung, S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562. MIT Press (2001)
12. Li, Y., Ngom, A.: Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data. In: BIBM, pp. 438–443. IEEE Press, Piscataway (2010)

13. Li, Y., Ngom, A.: A new kernel non-negative matrix factorization and its application in microarray data analysis. In: CIBCB, pp. 371–378. IEEE Press, Piscataway (2012)
14. Li, Y., Ngom, A.: The non-negative matrix factorization toolbox for biological data mining. BMC Source Code for Biology and Medicine 8, 10 (2013)
15. Li, Y., Ngom, A.: Sparse representation approaches for the classification of high-dimensional biological data. BMC Systems Biology (in press, 2013)
16. Ochs, M., Fertig, E.: Matrix factorization for transcriptional regulatory network inference. In: CIBCB, pp. 387–396. IEEE Press, Piscataway (2012)
17. Ochs, M., Rink, L., Tarn, C., Mburu, S., Taguchi, T., Eisenberg, B., Godwin, A.: Detection of treatment-induced changes in signaling pathways in sastrointestinal stromal tumors using transcripttomic data. Cancer Res. 69(23), 9125–9132 (2009)
18. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society - Series B: Statistical Methodology 67(2), 301–320 (2005)

A Local Structural Prediction Algorithm for RNA Triple Helix Structure

Bay-Yuan Hsu[1], Thomas K.F. Wong[2,⋆], Wing-Kai Hon[1], Xinyi Liu[3], Tak-Wah Lam[3], and Siu-Ming Yiu[3,⋆⋆]

[1] Department of Computer Science, National Tsing Hua University, Taiwan
[2] Ecosystem Sciences, CSIRO, Canberra, Australian Capital Territory, Australia
[3] Department of Computer Science, The University of Hong Kong, Hong Kong

Abstract. Secondary structure prediction (with or without pseudoknots) of an RNA molecule is a well-known problem in computational biology. Most of the existing algorithms have an assumption that each nucleotide can interact with at most one other nucleotide. This assumption is not valid for triple helix structure (a pseudoknotted structure with tertiary interactions). As these structures are found to be important in many biological processes, it is desirable to develop a prediction tool for these structures. We provide the first structural prediction algorithm to handle triple helix structures. Our algorithm runs in $O(n^3)$ time where n is the length of input RNA sequence. The accuracy of the prediction is reasonably high, with average sensitivity and specificity over 80% for base pairs, and over 70% for tertiary interactions.

1 Introduction

Prediction of a pseudoknotted secondary structure (base pairs crossing each other) of an RNA molecule is NP-hard in general [1]. In practice, the project focus on restricted classes of pseudoknots that are found in nature. Examples of these prediction algorithms include [1–7]. All these existing methods have an assumption that each nucleotide can interact with at most one nucleotide in the RNA. However, if tertiary interaction (where some single stranded nucleotides also form hydrogen bonds with nucleotides in base pairs) is considered, this assumption may not hold. Triple helix structure in ncRNA is a pseudoknotted structure with tertiary interaction. Figure 1 shows an example of a triple helix structure. Triple helix structures exist in yeast and human telomerase, and are found to be essential in quite a few biological processes (e.g. chromosome stability in stem cells, germline cells and cancer cells [8–10]; ribosomal frameshifting [11, 12]).

There are only two recent results [13, 14] that consider tertiary interactions. Siederdissen *et al.* provided a folding algorithm for RNA secondary structures which consider tertiary interactions inside only a regular structure (one without

⋆ B.Y. Hsu and T. Wong contributed equally.
⋆⋆ Corresponding author.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 102–113, 2013.

pseudoknots) while Wong *et al.* considered a structural alignment problem for RNA secondary structures with *standard triple helix structure* (tertiary interactions inside a simple pseudoknot). In this paper, we provide the first RNA secondary structure prediction algorithm for tertiary interactions over pseudoknots and focus on the standard triple helix structure defined in [14].

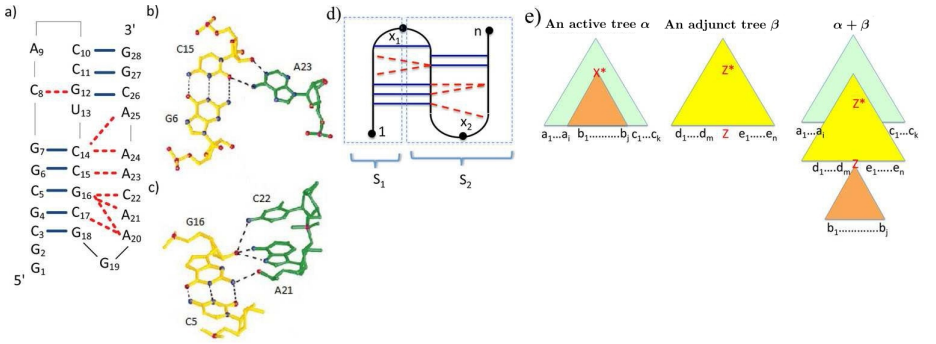


Fig. 1. (a) Triple helix in beet western yellows virus pseudoknot [11]. Blue lines represent the secondary structure. Red lines represent the tertiary interactions between single stranded nucleotides (according to the secondary structure) and base pairs. (b) and (c) Detailed view of some tertiary interactions in the structure [11]. (d) A standard triple helix structure. (e) Adjoining interaction between one active tree and one adjunct tree in simple tree adjoining grammar (STAG). The * indicates an active node. The active node X is replaced by the whole tree β.

We employ a machine learning approach (similar to the approach used by [15]) as follows to solve the problem. We define a grammar, which for any given RNA sequence, generate different possible secondary structures of the sequence. Based on some training datasets (the RNA sequences with known secondary structures), we assign probability to each grammar rule. Then, for each RNA sequence with unknown secondary structure, we can derive the optimal secondary structure (the one with the highest probability). Our contributions include the following. Existing grammars cannot handle standard triple helix structures. Based on the *simple tree adjoining grammar* (STAG) defined by [7] that can handle pseudoknots, we provide an extended version to cover the standard triple helix structures. Since STAG is an ambiguous grammar (i.e. there can be more than one derviation forming the same structure), we remove the ambiguity by introducing some restrictions on the grammar. Finally, we develop a dynamic programming algorithm that runs in $O(n^3)$ time, where n is the length of the input RNA sequence, to report the most probable structure[1] based on the probability measures. According to our experiments, the performance of our tool is

[1] The tool can be modified to report the top x possible structures, but for simplicity, we only consider the most probable structure in all our experiments.

reasonably good (with average sensitivity and specificity higher than 80% for base pairs and over 70% for tertiary interactions) when it is used for prediction of triple helix structures.

2 Standard Triple Helix

Based on [14], the formal definition of a standard triple helix is listed as follows. Let $A = a_1 a_2 \dots a_n$ be a length-n RNA sequence. Let M be the set of base pairs denoted as $M = \{(i,j) \mid 1 \leq i < j \leq n, (a_i, a_j) \text{ is a base pair}\}$. The tertiary interactions P of A are defined as follows. The interaction of the base pair (i,j) and the single stranded nucleotide k is denoted as $(i,j) * k$. That is, $P = \{(i,j) * k \mid (i,j) \in M, a_k \text{ is a single stranded nucleotide and interacts with } (a_i, a_j)\}$. Then, $H = (M, P)$ is referred as the triple helix structure of A.

The secondary structure still obeys the rule that no two base pairs share the same position. That is, for any $(i_1, j_1), (i_2, j_2) \in M$, $i_1 \neq j_2$, $i_2 \neq j_1$, and $i_1 = i_2$ if and only if $j_1 = j_2$. However, the tertiary interactions do not follow this rule, so that for any $(i_1, j_1) * k_1, (i_2, j_2) * k_2 \in P$, if $i_1 = i_2$ and $j_1 = j_2$, it does not imply $k_1 = k_2$; also, if $k_1 = k_2$, it does not imply $i_1 = i_2$ and $j_1 = j_2$. $H = (M, P)$ is a *standard triple helix structure*, as illustrated in Figure 1(d), if $\exists x_1, x_2 (1 \leq x_1 < x_2 \leq n)$, so that base pairs in M can be partitioned into two groups $R_1 = \{(i,j) \in M \mid 1 \leq i < x_1 \leq j < x_2\}$ and $R_2 = \{(i,j) \in M \mid x_1 \leq i < x_2 \leq j \leq n\}$, and H satisfies the following.

(1) For any two base pairs $(i_1, j_1), (i_2, j_2) \in R_k$, $k = 1$ or 2, either $i_1 < i_2 < j_2 < j_1$ or $i_2 < i_1 < j_1 < j_2$. That is, the base pairs in the same group do not cross. We say M forms a *simple pseudoknot* structure.
(2) For any $(i,j) * k \in P$, if $(i,j) \in R_1$, then $x_2 \leq k \leq n$ and $\nexists (i', j') \in R_2$ such that $j \leq i' \leq k \leq j'$ or $i' \leq j \leq j' \leq k$. This is to ensure that k is from a region outside that of R_1, and there does not exist base pairs in R_2 crossing with the tertiary interaction. Similarly, if $(i,j) \in R_2$, then $1 \leq k < x_1$ and $\nexists (i', j') \in R_1$ such that $k \leq i' \leq i \leq j'$ or $i' \leq k \leq j' \leq i$.
(3) For any $(i_1, j_1) * k_1, (i_2, j_2) * k_2 \in P$, if $(i_1, j_1), (i_2, j_2) \in R_1$, then $i_1 \leq i_2 \Leftrightarrow k_1 \leq k_2$, $i_2 \leq i_1 \Leftrightarrow k_2 \leq k_1$. This is to ensure that if the same single stranded nucleotide interacts with two base pairs, the interactions do not cross. Similarly, if $(i_1, j_1), (i_2, j_2) \in R_2$, then $j_1 \leq j_2 \Leftrightarrow k_1 \leq k_2$, $j_2 \leq j_1 \Leftrightarrow k_2 \leq k_1$.

3 Method

3.1 Simple Tree Adjoining Grammar

Simple Tree Adjoining Grammar (STAG) is a tree-based grammar for the generation of strings. The basic idea is to start with an initial tree, and then by repeatedly replacing some internal node of the current tree with another tree, bases or base pairs can simultaneously be added to the string that the tree represents. STAG can be used to predict pseudoknotted structures [7].

Let V be a finite set of alphabets and Σ be the terminal alphabet where $\Sigma \subset V$. Let γ be a tree over V such that (1) each internal node must be labeled with a nonterminal; (2) each leaf node can be labeled with a terminal or a nonterminal symbol; (3) each internal node can have any number of children; and (4) each node has a state, either active or inactive. A tree is *simple* and *active* if there is only one internal node active.

Let $Y(\gamma)$ (i.e. *yield* of a tree rooted at γ) be the string of labels of the leaf nodes of γ from top to bottom and from left to right. Precisely, it is defined as follows (let γ^1,γ^2,...,γ^n be the children of γ):

$$Y(\gamma) = \begin{cases} \text{label of } \gamma \text{ // if } \gamma \text{ is a leaf node} \\ Y(\gamma^1)Y(\gamma^2)...Y(\gamma^n) \text{ // otherwise} \end{cases}$$

In STAG, a tree β is an *adjunct tree* if: (1) there are only leaves labeled with nonterminal symbols; (2) there is only one internal node active; (3) the active internal node is along the backbone. The *backbone* is the path from the root to the leaf with nonterminal symbol.

A simple active tree α can be *adjoined* by an adjunct tree β and form a new tree denoted by $\alpha + \beta$. The adjoining interaction consists of the following operations (as shown in Figure 1e): (1) the active node in α is replaced by the tree β; and (2) the children of the active node in α become the children of the leaf with nonterminal symbol in β.

Definition 1. *$G(C, A)$ is defined as **Simple Tree Adjoining Grammar**, where C is a set of trees, all trees inside are simple and active, their yields are empty strings,and A is a set of adjunct trees.*

*A tree γ is a **derived** tree of G (where the set of the derived trees of G is denoted as $D(G)$) if either of the following constraints is satisfied (which is a recursive definition): (1) $\gamma = \alpha + \beta$ for $\alpha \in C$, $\beta \in A$. (2) $\gamma = d + \beta$ for $d \in D(G)$, $\beta \in B$.*

The language of G (denoted as $L(G)$) is defined as follows: $L(G) = \{w|w = Y(d)$ where $d \in D(G)\}$.

3.2 Structural Prediction for Triple Helix

To model the generation of an RNA with triple helix structure, we set the center tree and the adjunct trees as shown in Figure 2a. There is one center tree and nine adjunct trees. Every adjunct tree will contribute at least one base to the RNA sequence. More precisely, the trees TS1, TS2 and TS3 will produce a single base, while TP1 and TP2 will produce a base pair. Similarly, the trees TSH1 and TSH2 are for producing a single base which has tertiary interaction with an existing base pair, while the trees TPH1 and TPH2 are for producing a base pair which has tertiary interaction with a single base. In the following, we will describe these adjunct trees and how a triple helix structure is generated.

As shown in Figure 1e, the yield of an active tree can be viewed as the concatenation of three sequences: sequence S_1 (i.e. $a_1a_2...a_i$) which is from the

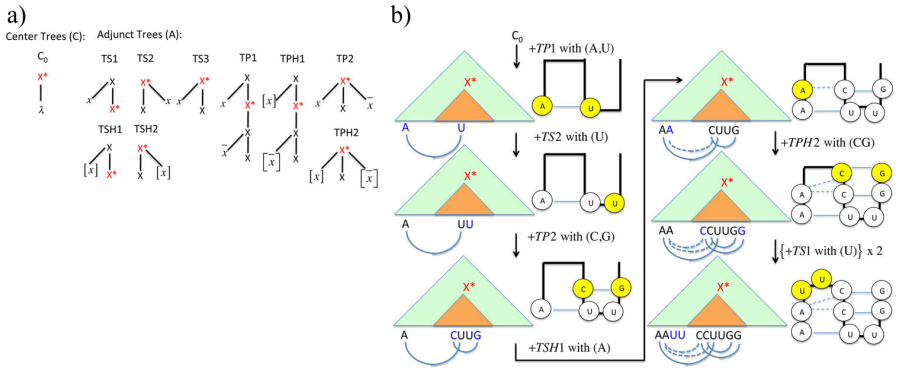


Fig. 2. a) The center tree and the adjunct trees of STAG for modeling RNA triple helix structure. The adjunct trees TS1, TS2 and TS3 generate single base. TP1 and TP2 generate base pairs. TSH1 and TSH2 generate single bases which interact with existing base pairs. TPH1 and TPH2 generate base pairs which interact with existing single bases. b) An example of generation of an RNA with triple helix structure by using STAG.

left part of the tree excluding the subtree of the active node; sequence S_2 (i.e. $b_1b_2...b_j$) which is from the subtree rooted at the active node; and sequence S_3 (i.e. $c_1c_2...c_k$) which is from the right part of the tree excluding the subtree of the active node. And by using the set of adjunct trees in Figure 2a, sequence S_3 is always an empty string, because none of the adjunct trees contribute any base to the sequence S_3. An RNA sequence can be viewed as the concatenation of S_1 and S_2 (as in Figure 1(d)), where S_1 represents the region $[1, x_1 - 1]$ while S_2 represents the regions $[x_1, n]$. The following lists out how the sequence S_1 and S_2 be modified when the tree is adjoined by a different adjunct tree. There are nine different operations (i.e one for each adjunct tree):

1. Adjoined by TS1: add a single base to the end of S_1.
2. Adjoined by TS2: add a single base to the end of S_2.
3. Adjoined by TS3: add a single base to the beginning of S_2.
4. Adjoined by TP1: add a base pair with bases at the end of S_1 and the beginning of S_2.
5. Adjoined by TP2: add a base pair with bases at the beginning and the end of S_2.
6. Adjoined by TSH1: add a single base at the end of S_1, which interacts with an existing base pair whose bases are at the beginning and the end of S_2, provided that the beginning and the end of S_2 are base pair.
7. Adjoined by TSH2: add a single base at the end of S_2, which interacts with an existing base pair whose bases are at the end of S_1 and the beginning of S_2, provided that the end of S_1 and the beginning of S_2 are base pair.
8. Adjoined by TPH1: add a base pair whose bases are at the end of S_1 and at the beginning of S_2, which interacts with the single base existing at the end of S_2, provided that the end of S_2 is a single base.

9. Adjoined by TPH2: add a base pair whose bases are at the beginning and the end of S_2, which interacts with the single base existing at the end of S_1, provided that the end of S_1 is a single base.

By using the above nine operations, it can build up any RNA with standard triple helix structure and any structure it comes up is a standard triple helix. An example of the generation of a standard triple helix is shown in Figure 2b. Under this model, different derivations may generate the same RNA sequence, but the corresponding secondary structures may be different. We associate probabilities for each tree operation (trained using real data); consequently, on given any input RNA sequence $A[1...n]$, we can report the derivation (and thus the corresponding secondary structure) that is the most probable.

To simplify the model, we assume that the probability of applying a particular tree p is independent of the current sequence, but depends on the previously applied tree p' and the bases involved in p. The probability of obtaining an input RNA sequence $A[1...n]$ with a particular secondary structure ζ is defined to be the product of the probabilities of the applied trees for operation in the corresponding derivation. To find out the most probable secondary structure ζ^* is equivalent to finding a ζ^* with the maximum summation of the log values of the corresponding derivation probabilities. Now, we define the following notations and present the recurrences.

- $M(i,j,k,p)$: the maximum score of the substructure $A[1...i] \cup A[j...k]$ of the sequence A if the last operation applied is p.
- $M_L(i,j,k,p)$: the maximum score of the substructure $A[1...i] \cup A[j...k]$ of the sequence A if the last operation applied is p and (i,j) is a base pair.
- $M_R(i,j,k,p)$: the maximum score of the substructure $A[1...i] \cup A[j...k]$ of the sequence A if the last operation applied is p and (j,k) is a base pair.
- $M_F(i,j,k,p)$: the maximum score of the substructure $A[1...i] \cup A[j...k]$ of the sequence A if the last operation applied is p and i is a single base.
- $M_G(i,j,k,p)$: the maximum score of the substructure $A[1...i] \cup A[j...k]$ of the sequence A if the last operation applied is p and k is a single base.
- $\texttt{score}(p,p',X)$: the score from previous operation p' to the current operation p with character set X. The scores are fixed in the parameter-tuning step of the method.
- $\texttt{charset}(i,j,k,p)$: the base(s) involved when the current operation p is applied.

$$M(i,j,k,p) = \max \begin{cases} M_L(i,j,k,p), M_R(i,j,k,p), M_F(i,j,k,p), M_G(i,j,k,p) \\ \text{// if } p \text{ is operation 3, also check the following score} \\ \max_{p'}\{M(i,j+1,k,p') + \texttt{score}(p,p',\texttt{charset}(i,j,k,p))\} \end{cases}$$

$$M_L(i,j,k,p) = \begin{cases} \text{// if } p \text{ is operation 1, 3, 5, 6 or 9} \\ -\infty \\ \text{// else if } p \text{ is operation 2 or 7} \\ \max_{p'}\{M_L(i,j,k-1,p') + \texttt{score}(p,p',\texttt{charset}(i,j,k,p))\} \\ \text{// else if } p \text{ is operation 4} \\ \max_{p'}\{M(i-1,j+1,k,p') + \texttt{score}(p,p',\texttt{charset}(i,j,k,p))\} \\ \text{// else if } p \text{ is operation 8} \\ \max_{p'}\{M_G(i-1,j+1,k,p') + \texttt{score}(p,p',\texttt{charset}(i,j,k,p))\} \end{cases}$$

The recurrence of $M_R(i,j,k,p)$ is analogous to that of $M_L(i,j,k,p)$. And the recurrence of $M_F(i,j,k,p)$ and $M_G(i,j,k,p)$ are similar too. The desired derivation corresponds to the entry among $M(i,i+1,n,p)$, for all possible i and p, that contains the maximum value. Once this entry is known, it is straightforward to obtain the corresponding secondary structure ζ^* by backtracking. By performing dynamic programming, each entry of $M(i,j,k,p)$, $M_L(i,j,k,p)$, $M_R(i,j,k,p)$, $M_F(i,j,k,p)$, and $M_G(i,j,k,p)$ can be computed in $O(1)$ time based on the previously computed entries. As there are altogether $O(n^3)$ entries to be filled, the time complexity of our prediction algorithm is $O(n^3)$.

A structural prediction grammar is ambiguous if there exists more than one derivation forming the same secondary structure, and [16] showed that an ambiguous grammar may not always report the optimal secondary structure correctly. The details of how the ambiguity of the grammar is removed is described in Appendix I. The accuracy of the prediction algorithm largely depends on how accurate the parameters $\texttt{score}(p,p',X)$ are. We only consider AU, UA, CG, GC, GU and UG as the possible base pairs and also regard the score for the operation with base pair AU (or CG or GU) is the same as that with base pair UA (or GC or UG). After considering all these together with the restrictions for preventing ambiguity, there are around 360 parameters $\texttt{score}(p,p',X)$ required to compute. We follow the maximum-likelihood approach mentioned by [17] to tune the grammar by a set of RNA sequences with known triple helix structures. $score(p,p',X)$ can be divided into two part: transition $a_{p'\to p}$ is score from previous operation p' to the current operation p, and emission $e_p(X)$ is score for X is involved in operation p. Since ambiguity is removed, operations series for each training sequence are known. We count the number of times each transition and emission, let these be $A_{p'\to p}$ and $E_p(X)$. Then the maximum likelihood estimators for $a_{p'\to p}$ and $e_p(X)$ are given by $a_{p'\to p} = \frac{A_{p'\to p}}{\sum_{l'} A_{p'\to l'}}$ and $e_p(X) = \frac{E_p(X)}{\sum_{X'} E_p(X')}$

With a set of training data, it takes O(n) time to calculate operation series for each sequence, and O(1) time to calculate all maximum likelihood estimators. For details, one may refer to [17].

4 Experimental Results

We implemented both the tuning and the prediction algorithms using C. There are three RNA families from Rfam 9.1 database with triple helix structures: RF00024, RF01050 and RF01074 (as listed in Table 1). The corresponding triple helix structure of each family can be deduced from [8, 9, 11, 18]. In the first experiment, we extracted the sequences of the triple helix regions of all the seed members (in Rfam 9.1 database, for each family, there is a set of reliable members that are regarded as seed members). It is found that the same model can hardly work well for the RNAs with large length difference. Since the lengths of the triple helix regions of the families RF00024 and RF01050 are similar, we put all the sequences from these two families together as set D_1, and the other sequences as set D_2.

Table 1. The families with triple helix structures

Family ID	# of seed members	Ave. length of triple helix region
RF00024	37	118
RF01050	13	99
RF01074	4	28

Table 2. Performance of triple helix prediction on the RNA sequences in set D_1 when using 10-fold cross-validation approach

Group	Base pairs		Tertiary interactions		Group	Base pairs		Tertiary interactions	
	Sen (%)	Spec (%)	Sen (%)	Spec (%)		Sen (%)	Spec (%)	Sen (%)	Spec (%)
G_1	90.5	89.6	76.2	88.9	G_6	95.1	90.1	78.3	78.3
G_2	81.3	77.1	72.2	76.5	G_7	56.6	65.8	44.4	38.1
G_3	97.1	90.3	88.9	88.9	G_8	90.8	90.1	95.7	88.0
G_4	79.5	76.5	38.5	71.4	G_9	92.5	87.5	71.4	74.1
G_5	86.8	87.5	73.9	77.3	G_{10}	74.5	82.0	64.5	66.7
On average: for base pairs, sensitivity **84.5** and specificity **83.7** for tertiary interactions, sensitivity **70.4** and specificity **74.8**									

We use the 10-fold cross-validation approach to evaluate the accuracy of our prediction tool. We evenly distributed all the sequences in the set D_1 into ten groups $G_1, G_2, \ldots, G_{10}$ such that the ratios of the sequences from each family are similar in each group. Next, we repeat the following procedure for each i from 1 to 10: We keep the group G_i aside, so that all the sequences as well as their corresponding triple helix structures from the other groups (i.e., $D_1 \backslash G_i$) were used for tuning our model; after that, the tuned model was used to predict the triple helix structure of each of the sequences in group G_i, and the predicted structure of each sequence was then compared with the corresponding real structure.

Our tool will report a set of base pairs as well as the tertiary interactions in the given region. Table 2 shows the summary of the performance of our prediction algorithm. Our method can predict the base pairs well with average sensitivity 84.5% and specificity 83.7%, and has a reasonable performance on the tertiary interaction prediction with over 70% in both sensitivity and specificity. Figure 3a shows an example of the predicted structure of the triple helix region of a sequence in family RF00024. The predicted structure is very similar to the real structure. Only one base pair (15,49) and one tertiary interaction (22,42)*74 are not predicted. Only one base pair (7,51) which should not exist is added.

For D_2, all RNA sequences are from the same family RF01074. The triple helix structures of the sequences are quite complex. There exist two or more single bases having tertiary interactions with the same base pair, and also two or more base pairs having tertiary interactions with the same single base. The tuned model may be over-fitted due to the small number of sequences in this set, but we still present the results here in order to show that our model is flexible enough to handle such a complex triple helix structure. We used 4-fold cross-validation technique in this set. For base pair prediction, the average sensitivity

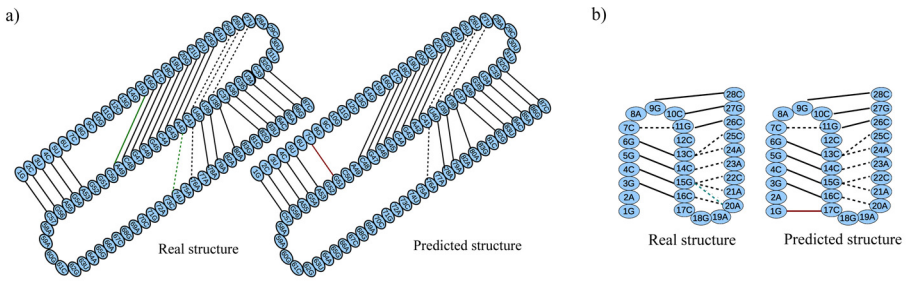


Fig. 3. a) Predicted triple helix region of sequence AF221906 of the family RF00024. b) Predicted triple helix region of sequence AF473561 of the family RF01074.

and specificity is 97.5% and 87.8%, respectively. For tertiary interaction prediction, the average sensitivity and specificity is 72.5% and 92.7%, respectively. Figure 3b shows an example of the predicted structure of the triple helix region for a sequence in the family RF01074. The predicted structure is very similar to the real structure, despite that the triple helix structure is quite complex.

Table 3. Experiment on the whole pipeline for the family RF00024

Seq ID	Annotated pseudoknot region	Reported pseudoknot region by vsfold5	Tertiary interaction predicted		Seq ID	Annotated pseudoknot region	Reported pseudoknot region by vsfold5	Tertiary interaction predicted	
			Sensitivity	Specificity				Sensitivity	Specificity
AF221911	55-143	7-173	80%	80%	AF221924	28-157	42-148	60%	60%
AF221913	63-148	52-148	100%	100%	AF221932	63-183	50-184	80%	80%
AF221916	19-139	2-165	50%	50%	AF221923	64-184	45-181	80%	80%
AF221926	55-138	51-172	80%	80%	AF221929	50-169	70-181	80%	80%
AF221940	56-135	45-169	100%	20%	AF221937	65-184	38-189	80%	80%
AF221934	60-155	48-184	100%	83%	AF221909	33-151	53-161	60%	60%
AF221927	60-156	31-152	80%	80%	AY058901	22-144	16-156	75%	75%
AF221910	62-151	74-197	100%	100%	AC121792	22-144	20-160	75%	50%
On average: Sensitivity **80%** Specificity **72%**									

In the second experiment, we try the whole pipeline for triple helix prediction on RNA sequences. Given an RNA sequence, the pseudoknotted structure will first be predicted by vsfold5 [19]. Then for those pseudoknotted regions reported by the tools, our tool predicts the triple helix structure. The maximum length of sequence vsfold5 supports is 450. The sequences in RF01050 are too long. Thus we selected those not-too-long sequences in RF00024 for the experiment. The pipeline is found to be feasible and quite effective (as shown in Table 3). On average, the sensitivity is 80%, while the specificity is 72%.

5 Discussion and Conclusions

To further evaluate our algorithm on the distinguishing power between regions containing a triple helix structure and those not containing one, we have selected

the families with simple pseudoknot structures (and no reported triple helix structures) as negative cases and it is found that our method can distinguish between regions with or without triple helix structure reasonably well. Since there are not much real data with known tertiary structures, further studies include collecting more real data, conducting a more comprehensive evaluation on the algorithm, and refining the grammar and the prediction algorithm to cater for more types of triple helix structures.

Acknowledgement. This project is partially supported by the General Research Fund (GRF) of the Hong Kong Government (HKU 719611E).

References

1. Lyngso, R., Pedersen, C.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. In: Proc. of the Fourth Annual International Conferences on Compututational Molecular Biology (RECOMB 2000). ACM Press (2000)
2. Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discrete Applied Mathematics 104, 45–62 (2000)
3. Chen, H., Condon, A., Jabbari, H.: An $O(n^5)$ algorithm for MFE prediction of kissing hairpins and 4- chains in nucleic acids. Journal of Computational Biology 16(6), 803–815 (2009)
4. Dirks, R., Pierce, N.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. Journal of Comput. Chem. 24(13), 1664–1677 (2003)
5. Reeder, J., Giegerich, R.: Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. BMC Bioinformatics 5, 104 (2004)
6. Rivas, E., Eddy, S.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. Journal of Molecular Biology 285(5), 2053–2068 (1999)
7. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree adjoining grammars for RNA structure prediction. Theoretical Computer Science 210, 277–303 (1999)
8. Qiao, F., Cech, T.R.: Triple-helix structure in telomerase RNA contributes to catalysis. Nature Structural and Molecular Biology 15(6), 634–640 (2008)
9. Chen, J.L., Greider, C.W.: Functional analysis of the pseudoknot structure in human telomerase RNA. Proc. Natl. Acad. Sci. USA 102, 8080–8085 (2005)
10. Theimer, C.A., Blois, C.A., Feigon, J.: Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. Molecular Cell 17, 671–682 (2005)
11. Su, L., Chen, L., Egli, M., Berger, J.M., Rich, A.: Minor groove RNA triplex in the crystal structure of a ribosomal frameshifting viral pseudoknot. Nature Structural Biology 6(3), 285–292 (1999)
12. Chen, X., Chamorro, M., Lee, S.I., Shen, L.X., Hines, J.V., Tinoco Jr., I., Varmus, H.E.: Structural and functional studies of retroviral RNA pseudoknots involved in ribosomal frameshifting: nucleotides at the junction of the two stems are important for efficient ribosomal frameshifting. EMBO 14(4), 842–852 (1995)

13. Siederdissen, C., Bernhart, S., Stadler, P., Hofacker, I.: A folding algorithm for extended RNA secondary structures. Bioinformatics 27, i29–i36 (2011) (ISMB 2011)
14. Wong, T.K., Lam, T., Yiu, S.: Structural alignment of RNA with triple helix structure. Journal of Computational Biology 19(4), 365–378 (2012)
15. Matsui, H., Sato, K., Sakakibara, Y.: Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures. Bioinformatics 21, 2611–2617 (2005)
16. Dowell, R.D., Eddy, S.R.: Evaluation of several lightweight stochastic context-free grammars for rna secondary structure prediction. BMC Bioinformatics 5, 71 (2004)
17. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Covariance models: SCFG-based RNA profiles. In: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1998)
18. Chastain, M., Tinoco, I.J.: A base-triple structural domain in RNA. Biochemistry 31, 12733–12741 (1992)
19. Dawson, W.K., Fujiwara, K., Kawai, G.: Prediction of RNA pseudoknots using heuristic modeling with mapping and sequential folding. PLoS One 2(9), e905 (2007)

Appendix I : Removing the Grammar Ambiguity

A structural prediction grammar is ambiguous if there exists more than one derivation forming the same secondary structure, and [16] showed that an ambiguous grammar often could not report the optimal secondary structure correctly. Therefore, we have to remove the ambiguity of the grammar such that each derivation can report a unique secondary structure.

First, there exist different operation series that come up with the same structure. For example, an operation 1 followed by an operation 2 would come up the same structure as an operation 2 followed by an operation 1. As shown in Figure 4a, we do not allow the operation series in right which produce the same structure as the operation series in left. Also, some operation sequences are not possible. For example, to perform operation 6, the beginning and the ending bases of B have to be a base pair. Therefore, it is not possible for an operation 3 followed by an operation 6 (because after operation 3, a single base will be added to the beginning of B). Figure 4b lists all of the cases.

Second, as one may notice, the positions of X_1 and X_2 may not be unique according to the definition in Section 2. In order to avoid the ambiguity, as shown in Figure 4c, we set the values of X_1 and X_2 as follows:

$$X_1 = \min\{\min j \mid (i,j) \in R_1, \min i \mid (i,j) \in R_2\}$$

$$X_2 = \max\{\max j \mid (i,j) \in R_1, \max i \mid (i,j) \in R_2\} + 1$$

where R_1 and R_2 are sets of base pairs defined in Section 2.

Since $S_1 = [0...X_1 - 1]$ and $S_2 = [X_1...n]$. Therefore, we have the following restrictions:

a)

The operation series which come up the same structure. (To avoid ambiguity, the operations in the right side of each equation are not allowed.)

$1 \to 2 \equiv 2 \to 1$	$2 \to 3 \equiv 3 \to 2$
$1 \to 3 \equiv 3 \to 1$	$4 \to 2 \equiv 2 \to 4$
$5 \to 1 \equiv 1 \to 5$	$4 \to 7 \equiv 2 \to 8$
$5 \to 6 \equiv 1 \to 9$	

b)

The followings are the impossible operation series.

$2 \to 6$	$8 \to 6$	$6 \to 7$	$9 \to 8$
$3 \to 6$	$1 \to 7$	$9 \to 7$	$4 \to 9$
$4 \to 6$	$3 \to 7$	$5 \to 8$	$7 \to 9$
$7 \to 6$	$5 \to 7$	$6 \to 8$	$8 \to 9$

c)

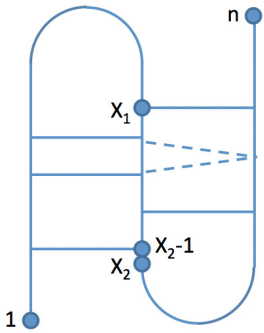


Fig. 4. a) The redundant operation series (in the right side of each equation). b) The impossible operation series. c) To remove the ambiguity, we define the exact values of X_1 and X_2.

1. S_2 cannot start with any single base. Since $3 \to 1$ and $3 \to 2$ are not allowed (see Figure 4a), we only need to restrict the operation 3 not being the last operation.

2. $X_2 - 1$ can be regarded as a center position of S_2 (which means all bases with positions $\leq X_2 - 1$ have to be added from the beginning of S_2, and all bases with positions $> X_2 - 1$ are added from the end of S_2) and the position $X_2 - 1$ cannot be a single base. There are two cases: the base $X_2 - 1$ belongs to a base pair $\in R_1$; or it belongs to a base pair $\in R_2$. In case 1, the operation 4 should be the first operation to add a base into S_2 and that position would be $X_2 - 1$. In case 2, there should be no operation 3 until the operation 5 or 9 exists. The left position of the base pair added would be $X_2 - 1$. According to the Figure 4a, since $2 \to 4$ and $2 \to 8$ are not allowed, therefore: we only need to restrict the operation 3 until the operation 4, 5 or 9 exists.

3. When R_1 and R_2 are empty, only operation 1 is allowed. i.e. When operations 4, 5, 8, 9 do not exist, only operation 1 can be the last operation.

4. If R_2 is not empty, R_1 has to be not empty too. i.e. If operation 5 or 9 exist, operation 4 or 8 has to exist before ends.

The above restrictions together with the restrictions listed in Figure 4 can make the grammar become unambiguous. Different derivation reports a unique secondary structure.

Combining Protein Fragment Feature-Based Resampling and Local Optimisation

Trent Higgs, Lukas Folkman, and Bela Stantic

Institute for Integrated and Intelligent Systems, Griffith University, Australia

Abstract. Protein structure prediction (PSP) suites can predict 'near-native' protein models. However, not always these predicted models are close to the native structure with enough precision to be useful for biologists. The literature to date demonstrates that one of the best techniques to predict 'near-native' protein models is to use a fragment-based search strategy. Another technique that can help refine protein models is local optimisation. Local optimisation algorithms use the gradient of the function being optimised to suggest which move will bring the function value closer to its local minimum. In this work we combine the concepts of structural refinement through feature-based resampling, fragment-based PSP, and local optimisation to create an algorithm that can create protein models that are closer to their native states. In experiments we demonstrated that our new method generates models that are close to their native conformations. For structures in the test set, it obtained an average RMSD of 5.09 Å and an average best TM-Score of 0.47 when no local optimisation was applied. However, by applying local optimisation to our algorithm, additional improvements were achieved.

1 Background

A fundamental aspect to modern molecular research is being able to elicit the three-dimensional structure of protein molecules. To date, there are roughly 20 million protein sequences stored in the UniProtKB/TrEMBL databases [1], but approximately only 79,000 of these sequences have available solved structures. Furthermore, it has been demonstrated that even a single amino acid substitution in a protein sequence may result in significant changes in protein stability and structure [2]. This makes it difficult for molecular and cell biologists who need the three-dimensional structure of proteins for their research. Due to so many proteins lacking solved structures, a lot of focus has been placed on improving and developing new computational *protein structure prediction* (PSP) methods.

Computational PSP methods have been historically broken up into three categories. In comparative modelling [3], evolutionary related homologous templates that have a high sequence similarity to the target sequence are identified. Then, the target and templates are aligned to form a three-dimensional structure of the target protein. Finally, this is completed by combining models for loop regions and other segments that do not align properly between the template and target. On the other hand, proteins that belong to different evolutionary classes can

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 114–125, 2013.

have similar structures too. Therefore, threading methods [4] have been developed to allow a query sequence to be mapped directly onto three-dimensional structures of solved proteins. The main motivation here is to recognise folds that are similar to the query even if no evolutionary relationship between the query and the template protein is present. Finally, the last category, *ab initio* [5], is used when the query sequence has no evolutionary related proteins in the template library. This is the most challenging approach, and success is at present limited to small proteins.

PSP has been tackled from numerous angles using one or more of the above methods. Some of the most successful approaches for *ab initio* are techniques that employ a fragment-based search strategy (e.g., Rosetta [6] or I-Tasser [7]). Fragments are derived from protein structures stored in the Protein Data Bank (PDB) based on the likelihood that a segment of the target protein chain will fold into a similar motif that already exists within a structure deposited in the PDB. This fragment-based approach has many benefits, for instance, by using fragments, we can approximate the populated areas of the local potential energy surface for the backbone of the protein structure. This stems from philosophy that when a protein is folding, the local structure will switch between numerous possible local conformations [8]. Therefore, each fragment can be considered a possible candidate for a conformation of the local sequence, which allows an energy function to be used that does not explicitly calculate the local interaction energy (the fragment selection method has already considered local interactions). This simplification is helpful in the PSP process because calculating the interaction energy assumes that a correct potential energy surface is known, which may not be the case. Finally, one of the main benefits of using a fragment-based approach is that we can easily move a protein from one topological isomer to another through a single fragment replacement. This ability can be looked at as moving a protein from one local minimum on the local physical energy surface to another, which is difficult to do in a more continuous based search method like molecular dynamics due to the computational complexity of such a move.

Another technique that has been applied to the PSP problem to help improve prediction accuracy is local optimisation. Local optimisation algorithms use the gradient of the function being optimised to determine which move will bring the function value closer to its local minimum. There are many different methods that have been proposed for this purpose [9]. For example, *linear minimisation* performs a single step based on the gradient, and after a number of recursive invocations, it reaches the local minimum. Compared to other available methods, it is considered rather slow. A variety of *quasi-Newton* methods were proposed in order to tackle local optimisation more efficiently. *Davidon-Fletcher-Powell* and *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) methods are such examples. In both cases, the descent's direction and step is computed according to the gradient and second derivatives of the function. The second derivatives are held in the form of *Hessian matrix* which can be efficiently updated. The extra information accumulated by these methods improves their efficiency, so that they converge faster. Furthermore, inexact search modifications of these methods have also

been proposed. They converge even faster, however, they do not necessarily reach the local minimum. Examples of these are *Armijo rule* and *non-monotone* modifications. In the latter case, the function value can be temporarily increased which may help escape shallow local minima. In another example, the *limited memory* variation of the BFGS method (L-BFGS) [10], instead of storing the whole Hessian matrix, only the vectors which represent the matrix implicitly are held in the memory.

Due to the success that fragment-based techniques have had, and the importance of local optimisation to keep every predicted model at the bottom of its energy basin, we combined both of these concepts to develop a PSP resampling approach that should be able to produce more accurate models. To achieve this, we carried out tests to identify which local optimiser performs the best and incorporated this optimiser into a *fragment feature-based resampling* approach which is discussed in more detail in the next section.

2 Methods

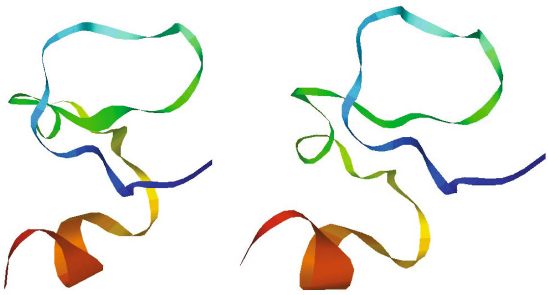
Local optimisation methods can be applied to the prediction process to guarantee that a PSP solution reaches the bottom of its energy basin. To determine the best local optimisation method for the PSP problem, we carried out tests utilising five state-of-the-art algorithms: *Linear Minimisation* (Lin-Min), *Broyden-Fletcher-Goldfarb-Shanno* (BFGS), *BFGS Armijo* (BFGS-A), *BFGS Armijo Non-monotone* (BFGS-A-NM), and *Limited Memory BFGS* (L-BFGS). To supplement these results and gauge the usefulness of local optimisation in the protein structure resampling process, the most promising algorithm was applied to our newly created *fragment feature-based resampling* approach. This new resampling algorithm builds on the concepts of our previous works [11–13].

In the next sections, our approach to analyse local optimisation techniques and our newly developed fragment feature-based resampling algorithm, which was designed to generate good starting points for local optimisation, are explained.

2.1 Local Optimisation

To identify which local optimisation methods perform well on the PSP problem, 128 native protein structures were selected and small random perturbations were applied to them in order to observe how successfully the local optimisers could guide these structures back to their native conformations. These native proteins structures were obtained from the CASP 8 website [14]. The *centroid energy function* [8] was chosen to be the objective function to be minimised using each of the five local optimisation methods (Lin-Min, BFGS, BFGS-A, BFGS-A-NM, L-BFGS). The same energy function was used for the implementation of our fragment feature-based resampling approach.

The general procedure used to test how well a local optimiser performed was by perturbing each structure by a certain amount of residues (between 1 and 3) and degree of movement (between 1 to 15 degrees), applying local optimisation,



(a) Original structure (b) Altered structure

Fig. 1. An example of a protein that had three of its residues perturbed by 15 degrees. Notice that the structure in (b) has several features displaced compared to its original structure in (a). All images were generated using Rasmol [16].

and then, evaluating how much the energy and structural similarity changed. The evaluation was carried out by recording the initial energy of the native structure, then recording the energy and root mean square deviation (RMSD) [15] of the altered structure, and finally, recording the energy and RMSD of the structure after local optimisation. The averages (across the set of all structures) of these values were then used as our final results. An example of one of our perturbations can be found in Figure 1.

2.2 Fragment Feature – Based Resampling

In our previous works on feature-based resampling using a genetic algorithm (GA) [11, 12], we demonstrated that by combining 'native-like' features generated from decoys from other PSP approaches, we could produce structures that were closer to the native conformations. To further this work, we created a *fragment feature-based resampling* algorithm to create 'near-native' starting points for local optimisation.

In our *GA feature-based resampling* algorithm [11, 12], our features were stored as the initial population in the form of decoys outputted from an initial prediction run. Then, crossover and mutation techniques were applied to them throughout the prediction process using energy function for fitness calculations. This was accomplished by using a crossover operator that splices together protein fragments that have 'native-like' features according to the fitness function f. Our GA's crossover operator randomly selected a crossover point (n) where $n \in C_\alpha(S)$ ($C_\alpha(S)$ refers to the set of C_α atoms contained within the structure S). Let $p1$ be parent 1, and $p2$ be parent 2. Everything from n onwards in $p1$ is replaced with everything from n onwards in $p2$, and vice versa. This process produced two offsprings.

In this work, we created a *fragment* feature-based resampling algorithm to overcome some of the limitations that were apparent from our results, the most

obvious being the inconsistencies of the energy function. The *centroid energy function* is not optimised to minimise its energy score in correlation with the RMSD of the structure being predicted, which has been discussed in our previous works [11, 12] and also shown in [17, 18]. This lack of accuracy can heavily affect the GA optimisation process as it relies on the energy function to guide it to more accurate solutions. To combat this, we developed an algorithm that incorporates *random* feature-sampling from a set of 'near-native' fragments.

Our algorithm works by taking a set of protein decoy structures and creating a fragment library from them. Each structure in the library can be broken into numerous fragments of different sizes. Sampling this space is then carried out by randomly selecting a position in the fragment library, randomly picking a fragment size (based on how much of the structure is left to put together), and finally, extracting that fragment based on the position of the structure being processed and the length of the fragment. There are two main constraints that our algorithm imposes on this fragment assembly procedure: (1) no structure can contain more than half of the residues of any given structure within the fragment library (to avoid duplicating any structure that was produced by the PSP suite), and (2) structures must have no collisions between residues.

The assembly process described above is run until 2,000 structures are generated. Based on our initial testing, we concluded that 2,000 structures is a sufficient amount of runs to generate most of the feasible combinations from the set of structures contained in our fragment library. As mentioned above, because we use an exhaustive search process, the energy function is only used to evaluate how well energy function can identify 'near-native' structures generated from our fragment feature-based resampling approach. Evaluation of the final output is carried out by two structural measures: RMSD [15] and template modelling score (TM-Score) [19].

3 Results and Discussion

We carried out two main tests: (1) assessment of which local optimiser performed the best in guiding structures back to their native conformations after random perturbation, and (2) evaluation of our fragment feature-based resampling algorithm with and without local optimisation. In the local optimiser test, 128 native proteins were randomly perturbed using the following criteria: 1 residue by 1 degree, 1 residue between 1–3 degrees, 2 residues between 1–5 degrees, 3 residues between 1–5 degrees, and 3 residues between 10–15 degrees.

For fragment feature-based experiment, the test set contained 14 protein structures. Our fragment library contained 1,000 structures for each prediction, and all fragments were generated from decoys. The local optimiser used for these tests was the one that performed the best in our first experiment. Each protein prediction was run five times, and the best output from each test was averaged for our final results to remove any bias caused by the random fragment assembly process. The best structure was chosen based on its RMSD value to its native conformation.

3.1 Empirical Results

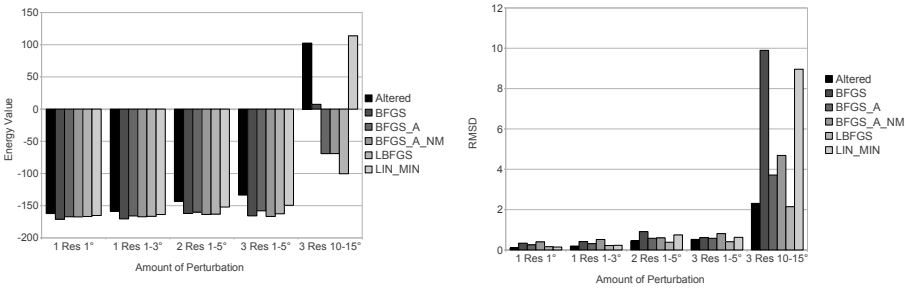
Figure 2 depicts the results that were gained from our perturbation experiments. The x axis is the amount of perturbation, and the y axis is the energy and RMSD values (Figure 2a and 2b, respectively). To complement these results, the local optimiser's ability to guide an altered structure back to its native conformation is visually demonstrated in Figure 3. Table 1 shows the results gained from our fragment feature-based approach. For each protein, the average best energy, RMSD, and TM-Score over the five tests with and without local optimisation are displayed. Finally, Figure 4 depicts the prediction ability of our fragment feature-based resampling by providing some visual comparisons between our models and their native conformations.

3.2 Analysis of Results

Local Optimiser Comparison. In our perturbation experiments, we used 128 protein structures, applied different amounts of perturbation to them, and then, locally optimised these structures. The average results for these experiments can be found in Figure 2. In Figure 2a, for the first four perturbation classes, it can be seen that all the local optimisers minimised the energy values starting from the altered structure. Also, in each of these cases every local optimiser achieved roughly the same energy levels after minimisation. For example, in Figure 2a, for the first perturbation class (1 residue with a perturbation of 1 degree), each optimiser generated models with an average energy between -165 and -171. However, in the last case (3 residues with a perturbation of 10–15 degrees), only BFGS-A, BFGS-A-NM, and L-BFGS minimised the energy significantly when compared to the average altered energy, with L-BFGS being the best. This suggests that the more a structure is altered from its native conformation, BFGS-A, BFGS-A-NM, and L-BFGS are more likely to guide it back to a stable state.

Other than just looking at the minimisation of the energy function to tell us which local optimiser performed the best, their ability to minimise the RMSD value of a structure was also evaluated. This would allow us to know which optimiser could lower the energy of a structure while also guiding it back to its native conformation. The results can be found in Figure 2b. From these results, it is clear that out of all the optimisers, only L-BFGS significantly guided altered structures back towards their native conformations. All the others had some success, but on average, they actually moved structures further away from their native state than the perturbation itself (this can be seen in Figure 2b where all the optimisers in every perturbation class, except L-BFGS, actually have worse RMSD averages when compared to the average altered RMSD).

Analysing the various perturbation classes in Figure 2b, it can be seen that even Lin-Min did well in minimising small perturbations (first two perturbation classes), however, as the structural change increased, its ability to move a structure back to its native state deteriorated, eventually becoming one of the worst out of the five we tested. It was also one of the worst optimisers at lowering the energy after a perturbation was made. L-BFGS, on the other hand,



(a) Energy (b) RMSD

Fig. 2. Results for our local optimiser comparison. In (a), the results of how well each local optimiser minimised the energy function are shown, and in (b), the results how well each optimiser performed in moving the altered structures back towards their native conformation are depicted. Note that these results are averaged from our complete 128 protein set, and the averages for the perturbed structures before local optimisation was applied are also included.

(a) Native (b) Altered (c) Optimised with L-BFGS

Fig. 3. Visual comparison of the native, altered and optimised structures. (a) is the native structure before perturbation, (b) is the altered structure, and (c) is the structure after L-BFGS optimisation was applied. As it can be seen in (c), once local optimisation was applied on the structure in (b), it moved back to its native structure. All images were generated using Rasmol [16].

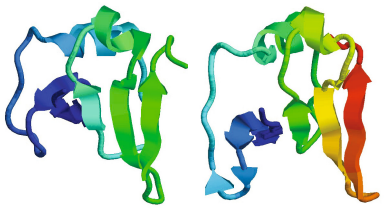
appears to always move the structure back towards its native state. From these findings, we can conclude that out of the five tested local optimisers, L-BFGS was most successful in regards to minimising the energy of structures after being perturbed while at the same time being able to guide the altered structures back towards their native states. To demonstrate the success of the L-BFGS optimiser, Figure 3 allows for a visual comparison of the native conformation, the perturbed structure, and the optimised structure using L-BFGS. It can be seen that the L-BFGS optimiser moved the altered structure back towards its native conformation by shifting the α-helices back into their correct places.

Fragment Feature-Based Resampling. After the perturbation experiment, we performed tests on our new fragment feature-based resampling approach, both with and without the L-BFGS optimiser. The results from these experiments can be found in Table 1. These results indicate that our new algorithm can resample features in such a way that on average 'near-native' models are generated. This is supported by an average best RMSD of 5.09 Å and an average best TM-Score of 0.48 when no local optimisation was applied. Another interesting aspect of these experiments is that the energy for the best scoring models (in terms of RMSD and TM-Score) have quite high energy scores. On average, they are not even in the negatives, meaning that the centroid energy function is rather limited in regards to finding structures that are low in RMSD. This is not to say that the centroid energy function is wrong as it has been proven that it works well in finding compact structures that are roughly close to their native states, but it lacks the accuracy to find models at a finer atomic resolution. A graphical representation of the predictive power of our fragment feature-based algorithm can be seen in Figure 4.

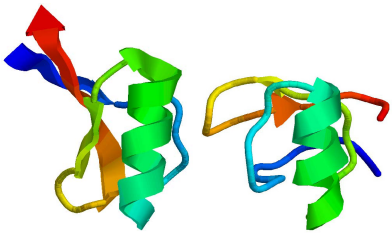
The next set of tests combined our algorithm with the L-BFGS local optimiser, which performed the best in our perturbation tests. In this experiment, we gained an average best RMSD of 5.05 Å and an average best TM-Score of 0.50. This

Table 1. Fragment feature-based resampling without and with local optimisation

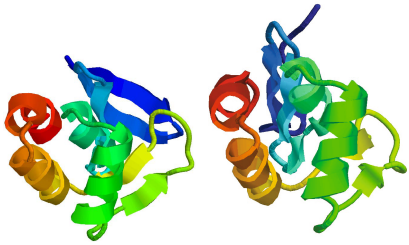
Protein	Without local optimisation			With local optimisation		
	f	***RMSD***	***TM-Score***	***f***	***RMSD***	***TM-Score***
79.1a91A	119.35	5.69 Å	0.37	142.73	5.70 Å	0.37
78.1aoyA	59.83	5.00 Å	0.55	43.26	4.99 Å	0.53
43.1bdsA	115.76	5.85 Å	0.23	89.52	5.61 Å	0.28
99.1bm8A	14.91	7.65 Å	0.29	62.96	7.62 Å	0.29
110.1brsABC	11.29	7.64 Å	0.50	42.45	7.74 Å	0.56
67.1cspA	12.20	2.95 Å	0.65	−18.66	2.75 Å	0.68
54.1enhA	65.25	5.13 Å	0.26	84.43	5.03 Å	0.28
76.1d3zA	−16.75	2.36 Å	0.76	−27.78	2.30 Å	0.76
47.1gptA	20.62	4.94 Å	0.38	75.19	5.03 Å	0.38
74.1kjsA	50.32	3.87 Å	0.55	32.37	3.91 Å	0.53
83.1pgxA	31.98	3.77 Å	0.66	−11.35	3.78 Å	0.66
77.1vccA	26.45	3.11 Å	0.67	12.81	3.19 Å	0.66
107.2pppA	164.93	8.57 Å	0.40	123.08	8.12 Å	0.49
78.2ptlA	39.38	4.70 Å	0.51	−14.94	4.89 Å	0.47



(a) 1vcc native (b) 1vcc predicted



(c) 1gpt native (d) 1gpt predicted



(e) 1bm8 native (f) 1bm8 predicted

Fig. 4. In (a), (c), and (e), the native conformations for proteins 1vccA, 1gptA, and 1bm8A, respectively, are depicted, and in (b), (d), and (f), the predicted models for these proteins using our fragment feature-based resampling algorithm are shown (note that local optimisation was not used on these structures). All images were generated using Rasmol [16].

means that irrespectively of the measure employed for the comparison, there were additional relative improvements (0.8% and 4.2% in the case of RMSD and TM-Score, respectively). The main reason why local optimisation in this case did not result in higher improvements was that the fragments were obtained from decoys which had already been locally optimised. However, if the algorithm was designed to fold protein structures from just the amino acid sequence, local optimisation would definitely be more useful.

There are aspects to our fragment feature-based approach that could be addressed to obtain further improvements. The first one is the problem of missing features in the fragment library. As features generated by other PSP suites are used in our approach, if the initial decoys do not contain all features necessary

to create the native conformation for a given protein, then, our algorithm will produce poor results. In most cases, given our results, nearly all features were present, however, an example of this problem occurring can be seen in Figures 4c and 4d. In Figure 4d, one of the major β-sheets was predicted incorrectly and also has the wrong orientation, which brings us to the other problem: the orientation of features.

Our approach stitches fragments together until the end of the protein chain is reached, however, it never takes into consideration the orientation these features should have. Figure 4 illustrates that some of the major reasons why we did not obtain better results was due to the orientation of the features. To combat this problem, we could add a move set that rotates the fragments around until their optimal placements are found. This brings up two challenges: firstly, a scoring function that can inform us what the best orientation is for a fragment or a set of fragments, and secondly, how much rotation should be applied. According to the literature, once a compact structure has been obtained it is best to only move fragments slightly (e.g., 1–5 degrees) [8]. If both of these problems were addressed, our algorithm could generate even better models than it already had.

4 Conclusions

Fragment-based protein structure prediction methods have shown a lot of success in predicting the three-dimensional conformations of proteins. In this paper, we combined fragment-based approach and local optimisation techniques. By doing this, we showed that our new *fragment feature-based resampling* algorithm can generate protein models close to native structures. Furthermore, we described the benefits and disadvantages of using local optimisation techniques in conjunction with feature-based resamplig.

To identify which local optimisation methods performed well on the PSP problem, we selected 128 native protein structures to which we applied small random perturbations in order to observe how successfully local optimisation could guide structures back to their native conformations. The five optimisers we tested were: *linear minimisation* (Lin-Min), *Broyden-Fletcher-Goldfarb-Shanno* (BFGS), *BFGS Armijo* (BFGS-A), *BFGS Armijo non-monotone* (BFGS-A-NM), and *limited memory BFGS* (L-BFGS). To supplement these results and gauge the usefulness of local optimisation in the protein structure resampling process, we took the most promising method from our perturbation experiment and combined it with a fragment feature-based resampling approach, which we proposed in this work.

Our new fragment feature-based resampling algorithm works by creating a fragment library from a set of protein decoys. Each structure in the library can be broken up into numerous sized fragments to build up 'near-native' protein models. Sampling this space is carried out by randomly combining fragments together until 2,000 collision-free structures are produced.

From our experimentation, we observed that the L-BFGS optimiser performed the best. It was able to both minimise the energy of a structure and bring a

structure back towards its native state. In regards to our fragment feature-based resampling algorithm, we demonstrated that it could generate 'near-native' models. Out of the 14 structures we tried to predict, it obtained an average best RMSD of 5.09 Å and an average best TM-Score of 0.47 when no local optimisation was applied. When we applied local optimisation, additional improvements in both RMSD and TM-Score were recorded.

As mentioned in our results discussion and analysis, there is two avenues to further improve our algorithm. First, being able to ensure that all features which are needed to generate the native conformation are present in the fragment library. However, this may be in some cases rather difficult as we are unsure what features the native model contains, but the probability could be increased if there is a sufficiently large library. And second, finding the correct orientation of the fragments is crucial to allow more accurate models to be produced.

References

1. Consortium, U.: The universal protein resource (uniprot) 2009. Nucleic Acids Research 37, D169–D174 (2009)
2. Folkman, L., Stantic, B., Sattar, A.: Sequence-only evolutionary and predicted structural features for the prediction of stability changes in protein mutants. BMC Bioinformatics 14(suppl. 2), S6 (2013)
3. Sali, A., Blundell, T.: Comparative protein modelling by satisfaction of spatial restraints. Journal of Molecular Biology 234(3), 779–815 (1993)
4. Zhang, Y., Skolnick, J.: Automated structure prediction of weakly homologous proteins on a genomic scale. PNAS 101(20), 7594–7599 (2004)
5. Simons, K.: et al. Prospects for ab initio protein structural genomics. Journal of Moleculer Biology 306, 1191–1199 (2001)
6. Meredith, D.: Rosetta tackles the extreme origami of protein folding. HHMI Bulletin 14, 20–23 (2001)
7. Zhang, Y.: Template-based modeling and free modeling by I-TASSER in CASP7. Proteins 8, 108–117 (2007)
8. Rohl, C., Strauss, C., Baker, D.: Protein structure prediction using rosetta. Methods Enzymology 383, 66–93 (2004)
9. Bonnans, J.: Numerical optimization: theoretical and practical aspects, 2nd edn. Springer (2006)
10. Liu, D., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical programming 45(1), 503–528 (1989)
11. Higgs, T., Stantic, B., Hoque, T., Sattar, A.: Genetic algorithm feature-based resampling for protein structure prediction. In: IEEE World Congress on Computational Intelligence, pp. 2665–2672 (2010)
12. Higgs, T., Stantic, B., Hoque, T., Sattar, A.: Refining genetic algorithm twin removal for high-resolution protein structure prediction. In: IEEE Congress on Evolutionary Computation CEC 2012, 251–258 (2012)
13. Folkman, L., Pullan, W., Stantic, B.: Generic parallel genetic algorithm framework for protein optimisation. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (eds.) ICA3PP 2011, Part II. LNCS, vol. 7017, pp. 64–73. Springer, Heidelberg (2011)
14. CASP8: 8th community wide experiment on the critical assessment of techniques for protein structure prediction (2008), `http://predictioncenter.org/casp8/` (last accessed: July 2012)

15. Carugo, O.: Statistical validation of the rootmeansquaredistance, a measure of protein structural proximity. Protein Engineering, Design and Selection 20(1), 3338 (2007)
16. Sayle, R.: Molecular visualization freeware and rasmol classic site (2009), `http://www.umass.edu/microbio/rasmol/index2.htm` (last accessed: February 2011)
17. Bowman, G., Pande, V.: Simulated tempering yields insight into the low-resolution rosetta scoring functions. Proteins: Structure, Function, and Bioinformatics 74, 777–788 (2009)
18. Shmygelska, A., Levitt, M.: Generalized ensemble methods for de nova structure prediction. PNAS 106(5), 1415–1420 (2009)
19. Zhang, Y., Skolnick, J.: Scoring function for automated assessment of protein structure template quality. Proteins 57, 702–710 (2004)

Experimental Determination of Intrinsic *Drosophila* Embryo Coordinates by Evolutionary Computation

Alexander V. Spirov[1,2,3,*], Carlos E. Vanario-Alonso[3], Ekaterina N. Spirova[2,3], and David M. Holloway[4]

[1] Computer Science and Center of Excellence in Wireless and Information Technology, Stony Brook University, NY, USA
[2] The I.M.Sechenov Institute of Evolutionary Physiology & Biochemistry, St.-Petersburg, Russia
[3] Applied Mathematics and Statistics and Center for Developmental Genetics, Stony Brook University, NY, USA
[4] Mathematics Department, British Columbia Institute of Technology, Burnaby, B.C. Canada

Abstract. Early fruit fly embryo development begins with the formation of a chemical blueprint that guides cellular movements and the development of organs and tissues. This blueprint sets the intrinsic spatial coordinates of the embryo. The coordinates are curvilinear from the start, becoming more curvilinear as cells start coherent movements several hours into development. This dynamic aspect of the curvature is an important characteristic of early embryogenesis: characterizing it is crucial for quantitative analysis and dynamic modeling of development. This presents a number of methodological problems for the elastic deformation of 3D and 4D data from confocal microscopy, to standardize images and follow temporal changes. The parameter searches for these deformations present hard optimization problems. Here we describe our evolutionary computation approaches to these problems. We outline some of the immediate applications of these techniques to crucial problems in *Drosophila* developmental biology.

1 Introduction

The completion of many genomic projects in the last decade has given rise to a new scientific objective, that of functional genomics - the next step towards the ultimate goal of a detailed understanding of how genome works [1].One of the critical questions in development is how the correct set of genes is expressed in each cell in order to form differentiated tissues. Research in *Drosophila* is reaching a stage where the expression of multiple genes can be followed dynamically in early embryogenesis at single cell resolution, in order to begin to understand the regulation underlying spatial patterning [2,3]. For instance, the BDTNP project [2] has currently mapped the expression of about 100 genes in each of about six thousand nuclei in early stage embryos; but these are initial steps of a very challenging project to trace as many related

* Corresponding author.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 126–137, 2013.

genes in individual development as possible, for as long and in as much detail as possible.

In *Drosophila*, the impressive experimental progress comes with unique data challenges. For instance, major challenges arise in mapping gene expression in early *Drosophila* development. The information comes from confocal microscopy scans [4], which present unique challenges for preprocessing, processing and analyzing sets and stacks of images. In this publication we will concentrate on computationally hard optimization problems in multidimensional confocal imaging of *Drosophila* embryos.

Data from large numbers of embryos must be combined to create data atlases from multiple genes and at multiple stages of development. Single embryos (fixed & stained) can be imaged for a few (usually three) segmentation genes. Therefore, data sets integrated from multiple embryos, stained for the variety of segmentation genes and over the entire patterning period, are necessary for gaining a complete picture of developmental dynamics. Images from individual embryos must be standardized to create such integrated data sets. Numerous sources of variability between images present challenges for data processing. These sources are both experimental and intrinsic to the biochemistry and biophysics of the developing embryos. Processing techniques which can separate experimental sources of variability allow for quantitation of the biological variability between embryos.

The standardization of multiple images is in essence a transformation of diverse sets of data into a single coordinate system; it is a general problem in medical and biological imaging. In *Drosophila*, major challenges arise from the different shapes and sizes of embryos, and the intrinsic curvilinearity of the chemical gradients specifying cell type. Intrinsic biological variability affects these factors, as do experimental treatments for data acquisition.

Standardization problems for *Drosophila* embryo images have been approached for 1D (gene expression profiles [5, 6, 7]), 2D (expression surfaces [8, 9]), and even 3D data [2, 10]. These approaches have involved elastic (or non-rigid) deformation of images to a single coordinate system [5, 6], which involve heavy use of computational resources. 3D views of the data are impressive and informative, but many statistical analyses and modeling projects are done in 1D or 2D; methods for reducing dimensionality are needed for data validation of such theoretical projects, and elastic deformation can also be used for this.

We have developed a type of elastic deformation for *Drosophila* analysis, following biometric coordinate transformations [5,6,11,12] first pioneered by D'Arcy Thompson [13], and used this for systematic studies of within- and between-embryo noise in 1D and 2D gene expression data [6, 14]. The approach has been adopted more recently by other teams [15, 16, 17].

In recent years, more and more laboratories are studying large sets of confocal images of early *Drosophila* embryos. Web bases include: FlyEx [18], which we have been involved with; the large-scale 3D BDTNP project (BID) [2]; and FlyFISH [19]. Similar datasets are under study in other labs [20,21,22,23,24]. All workers in this area face image processing challenges in extracting reliable information from confocal data. In this communication, we discuss the challenges presented in these types of datasets, present our approach to some of these fundamental problems, and report on new techniques we are developing, especially for application to new methods of data acquisition and to optimize processing.

2 Data and Nature of the Problems

In the first 4 hours of development, the major axes of the *Drosophila* embryo are established by gradients of gene expression products specifying particular cell fates in precise locations. The major, anterior-posterior (AP), axis is established by the segmentation network, a set of some 15-20 genes that establishes the striped patterns of gene expression which precede the anatomical appearance of the segmented body plan. This system has been intensively studied as a model for the functional genomics of spatial patterning [25, 26]. Figure 1 shows these striped ('pair-rule' gene) patterns. There are also chemical patterning gradients in the dorsal-ventral (DV) axis, orthogonal to the AP system. The intersection of these two systems establishes a coordinate system for the early embryo. Numerous cell types and structures have been shown to differentiate at particular intersection values of the AP and DV axes, for instance: the salivary glands, localized AP by a narrow band of *scr* gene expression and DV by the *dpp* gene [27]; neural cells differentiating at the intersection of achaete-scute gene patterns [28]; or structures developed at the intersection of *wg* and *sog* expression. These positions can be manipulated experimentally, such as by mutation. This intrinsic coordinate system is curvilinear, as seen by the bending of stripes in Fig. 1. The patterns become more curved with time. While patterning can be described in these intrinsic coordinates, standardization of images and subsequent analysis is aided by use of standardized coordinate systems, such as confocal elliptical or Cartesian. This communication presents techniques for transforming the embryo's intrinsic coordinates into a standard one.

2.1 Flattened vs. Intact Embryos

The quantitative data on segmentation genes are generally of two types, each presenting challenges to data analysis. These are 1) from confocal scans of flattened embryos, squeezed under a cover glass (Fig. 1A), and 2) from complete 3D scanning of physically intact embryos [29] (See Fig. 2). Gene expression datasets on flattened embryos are available on the FlyEx (protein) and FlyFISH (mRNA) web bases [30, 31]. (Data is more frequently taken in this way, and newer published data is also available from authors upon request.) 3D reconstructions of intact embryos are available on the BDTNP web base [32, 33].

These two approaches each have their advantages and disadvantages. Scanning of flattened embryos allows for a single focal plane, and is the most common, used in such databases as FlyEx. There are a number of methodological pitfalls with this approach, however, which must be addressed in the processing of such data. The chief problem is from the nearly arbitrary orientation of embryos under the cover glass. As an analogy, the problem is similar to placing a bunch of soft toy Rugby balls on a table and pressing them down with a sheet of glass. The lacing on the balls is analogous to the pair-rule stripes on the embryo. Not only will the laces curve as pressure is applied, different balls will have their laces oriented in different directions. This squeezing problem does not apply to intact embryo 3D reconstructions, so comparison of flattened 2D to intact 3D datasets first requires correction of the effects of the cover glass.

Datasets taken by each method offer different information - e.g. FlyEx has protein data and BDTNP has mRNA data – so it is desirable to be able to map between the flattened and intact data. In addition to correcting for the effects of flattening, this requires finding common landmarks (or ground control points), with the following challenges: 1) for flattened embryos we can observe slightly less than half of the nuclei (half of the cylindrical unwrapping; cf Fig. 1A & B); 2) if we superpose a flattened scan on the cylindrical projection of an intact embryo, the exact position of one image against another will be to some degree arbitrary.

An evident landmark in the 3D images is the dorsal axis of symmetry (see Fig. 1), where the stripes are closest to one another and locally perpendicular to the line of symmetry. The position of this line can be estimated in some 2D images (such as Fig. 1A), aiding alignment, but this is not a general property for all images. Adding to the alignment challenge is the curvature of the stripes. Part of this curvature is due to the intrinsic biological coordinates. But flattened images have an additional (and poorly controlled) curvature imposed from the experimental method.

2.2 Coordinate Transformation

Accounting for experimental effects on pattern and the steps to standardize the intrinsic curvilinear coordinates of embryos can be seen as problems in coordinate transformation. Correcting for experimentally-induced curvature (from embryo flattening) is a first step in data processing. Since intrinsic curvature varies between embryos, this too must be corrected to standardize multiple images. One approach to this standardization is to transform the curvilinear coordinates into a rectilinear Cartesian system. In one of the first works to investigate elastic coordinate transformations with respect to body plans, D'Arcy Thompson [13] made a classic deformation from a sunfish in curvilinear coordinates to a puffer fish in Cartesian coordinates. A similar transformation applies to the problem of standardization via stripe straightening in *Drosophila*. It took some 60 years after Thompson's graphical demonstrations for techniques to be formalized so that such transformations could be automated: in the 'bioorthogonal analysis' of Bookstein [34]; and in Siegel's [35, 36] technique for aligning and comparing homologous sets of landmark-coordinates. Morphometric coordinate transformations have expanded greatly in 30 years [37], for instance being applied in 2D structures such as insect wings [38]. We have developed a number of techniques in this area for application to *Drosophila* image processing [5, 6, 11, 12]. Stripe straightening is a major tool for standardizing images, which can be followed by registration of stripes for integrated data sets. Stripe-straightened data also provides dimensional reduction, producing data for verification of models and statistical analyses focused on 1D AP patterning. In addition, we have used the approach to standardize image intensity within and between images [7].

While stripe straightening focuses on the AP coordinate, there is also curvilinearity in the DV direction, especially for ventral positions. (For the intrinsic coordinates, it is known that DV morphogenetic gradients affect AP organization [see 39].) This two-dimensional curvilinearity is illustrated in the right hand images of Fig. 2. This secondary curvature can become a serious obstacle for automated data processing. Again, this may reduce to a coordinate transformation problem, if the intrinsic AP and DV curvature can be properly captured and transformed into a rectilinear system.

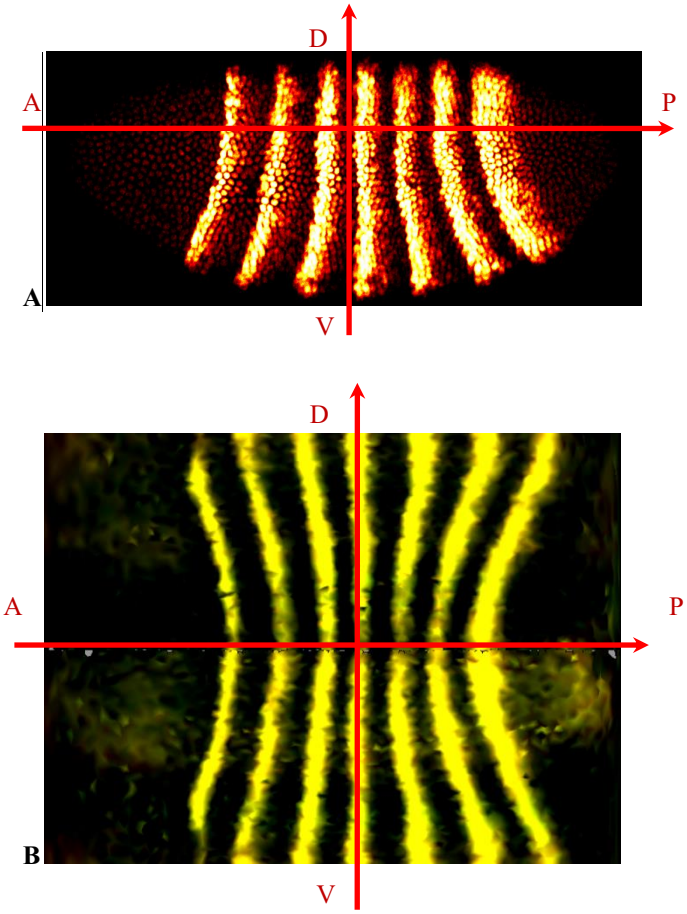


Fig. 1. The challenge of finding landmarks to juxtapose patterns from flattened and intact embryos. The two orthogonal axes of the striped pattern (red: y-axis along straightest stripe; vertical displacement of x-axis chosen to be most orthogonal to other stripes) tend to be invariant between the two approaches. (A) Image of flattened embryo with crescent-like stripes of expression of the pair-rule gene *eve*. (B) Unrolled (cylindrical projection) *eve* pattern for an intact embryo (3D reconstruction), with the same two orthogonal axes.

Intrinsic curvature also increases during development, especially as cells begin to move at the onset of the gastrulation stage. This change in geometry is important to study in its own right, as well as needing quantification for standardization of confocal data. The increasing curvature can be considered as an extension of the elastic deformation between Cartesian and curvilinear coordinates.

Computing such coordinate transformations is challenging: in addition to the wide range of intrinsic biological, experimental, and instrumental/observational sources of variability, there are no defined or standard reference solutions for such computations. Evaluating pattern coincidence between pairs of embryos at single cell resolution (at a

stage when embryos have ~6000 cells) can involve heavy, non-standard computation. Such problems can be well suited to evolutionary optimization; we have tested and developed a number of Genetic Algorithms (GA) approaches for this (see [5,6,11,12] & next section).

3 Techniques

Our coordinate transformations are based on optimization of polynomial maps between coordinate systems.

3.1 Stripe Straightening

The stripe straightening procedure is a transformation of the AP (x) coordinate by a polynomial of the form:

$$x' = Axy^2 + Bx^2y + Cxy^3 + Dx^2y^2 + \ldots \quad (1)$$

where $x = w - w^0$ and $y = -h - h^0$, and w and h are the initial spatial coordinates (AP and DV, respectively). The y-coordinate remains the same, while the x-coordinate is transformed as a function of both coordinates w and h (for details see [5, 6, 11, 12]). The exact form of (1) must be determined (more below), and the parameters w^0, h^0, A, B, C, D, etc. for each image must be found by an optimization technique. We tested a number of methods: GA; simplex; and a hybrid of these [5, 6, 11, 12]. We found GA to be the best for solving problems like (1) (especially for the multi-quadrant fitting discussed below). For GA optimization, we subdivide the image into a series of longitudinal strips. Each strip is subdivided into bins and the mean brightness (local fluorescence level) is calculated for each bin. Each row of means gives a profile of local brightness along each strip. A cost function is computed by pair-wise comparison of all profiles, summing the squared differences between bins. The task of the GA procedure is to minimize the cost function. The smaller the summed differences between strips, the closer the process is to the straightened endpoint. There is a possibility of over-straightening: this can be compensated by applying a penalty to any solution (parameter set) that moves more than one nucleus position past a predefined threshold (having a defined endpoint of straightened stripes helps here), though the penalty can influence search efficiency.

Intuitively, one can think of the straightening process as shrinking the image in such a way that the farther a given nucleus is from the dorsal edge and horizontal midpoint, the farther it must be moved towards the horizontal midpoint. More formally, we assume that the center of a pair-rule stripe follows a curve of constant AP position. The origin of the image coordinate system is at the top left, with image coordinates for width w increasing to the right and height h increasing down. To begin determining the final (straightened) AP and DV coordinates, x' and y' respectively, we note that there is an AP position (near mid-embryo) at which one stripe is vertical for its whole length. The center of this stripe defines x'=0 (the y'-axis). Each pair-rule stripe other than the one at x'=0 is curved; we vertically shift the x' axis so that it intersects each of the stripes at the point where it is vertical. Because of the vertical stripe: 1) the y- and y'-axes coincide; and 2) lines of y' = const are orthogonal

to the y- and y'-axes. The new coordinate system (x',y') has the same orientation and w0, h0 origin as the (x,y) system.

Analysis of the series has allowed us to eliminate all but three terms from the series [5, 6], so now we write an initial model of image transformation as

$$x' = x + Axy^2 + Bx^2 y + Cx^3 \tag{2}$$

All of these terms have a clear interpretation. The xy^2 term is the main one: it represents the quadratic DV curvature that increases with distance from the x-axis. The x^2y term gives the residual DV asymmetry and the x^3 term gives the residual AP asymmetry. Finally, expressing (2) in terms of w and h, gives

$$x' = w - w^0 + A(w - w^0)(h - h^0)^2 + B(w - w^0)^2(h - h^0) + C(w - w^0)^3 \tag{3}$$

In tests with this initial model, however, we found that in more than half of the cases it was insufficient for straightening stripes. Therefore, we expanded the model empirically, adding 4th order terms.

For performance on confocal images, we found the best polynomial to be

$$A + Bxy + Cxy^2 + Dx^2y + Ex^2y^2 + Fx^3y + Gxy^3 \tag{4}$$

We can understand some these additional fourth order terms as follows: Cx^2y^2 is a correction term for parabolic bending, while Dxy^3 serves to correct DV asymmetry. In general, the situation is typical of a polynomial approximation problem, where one polynomial is best but many others are very good.

We have found some independence in the stripe curvature between head and tail ends of the embryo, perhaps reflecting differences in underlying patterning mechanisms. This affects the straightening process, and we have found improved fitting by independent elastic deformations on the head and tail halves of the image [5, 6]. Test computations indicate independent deformation on the four quadrants of the image may be best, to also account for DV dependencies in stripe curvature. A full optimization can operate, therefore, on 3 quadrants times 7 parameters in eq. (4) for a total of 28 parameters (plus an evaluation of values w^0, h^0).

With sufficient data on DV patterning (currently only available on the BDTNP Web base, [29]), we can also apply an elastic deformation to straighten in DV. We have applied DV straightening after AP straightening, and found a third order polynomial (Cf with (4)) gives good results:

$$x' = h - h^0 + A(h - h^0)(w - w^0)^2 + B(h - h^0)^2(w - w^0) + C(h - h^0)^3 \tag{5}$$

(in terms of w and h). The DV procedure is generally less precise than for the seven-striped AP patterns.

To summarize, stripe-straightening has a number of steps and challenges, including: finding the exact form of the deformation polynomial; finding efficient optimization algorithms for this task; limiting over-deformation; using multi-strip and multi-sector (i.e. quadrant) optimization; and the complicated and variable 3D

geometry, including the squeezing effects of flattened images, which can affect the efficiency of the evolutionary computations.

3.2 Implementation

We have developed a set of computational tools to process 2D data for about ~3,000 (flattened) or ~6,000 nuclei (intact embryo). The tool uses ASCII files for individual embryos in the format of the FlyEx Web base. We also developed a script to convert PointCloud data files from the BDTNP Web base. The main function is the GA search for parameters of the elastic deformation (stripe straightening). The software includes a C++ version for Windows, a Delphi version (Windows), and a Free Pascal version for Linux/Unix. For each input file the software produces two output files: one with the straightened data (in the input data format); and one with the polynomial coefficients for the deformation. The software is available from the authors upon request.

4 Biologically Significant Results and Discussion

The spatial patterns we have presented here are created by genetic regulation, the extremely complex and at best partially understood system of interactions between gene products (and other factors). A number of theoretical models have been developed for the AP patterning system to characterize these interactions. Many of these models are developed in 1D, so the dimensional reduction discussed above, with stripe-straightening, serves as an important tool for data processing to validate models. The quantification of variability arising from the coordinate transformation also sheds light on other biological questions. We present a few examples of the biological application of our work here.

A fundamental question in development is how spatial expression patterns can develop precisely and reliably, despite operating at low concentrations which are associated with high noise. Many investigators are working on quantifying this intrinsic biochemical noise, and studying how it is reduced in order to produce embryonic patterns of the required precision.

The natural variability in stripe curvature between embryos also reflects variability in developmental conditions. Two embryos of the same age class can show large qualitative differences in this respect. Fig. 2 shows the middle (fourth) eve stripe curving to the right in one embryo and to the left in another. Quantification of curvature via the stripe-straightening transformation can allow for a deeper investigation of these effects; for instance studying the correlation between stripe bending and embryonic geometry.

In addition to noise and variability in gene expression, there is significant variability in cellular order. This variability increases as the embryo becomes cellularized and begins the process of gastrulation. This variability is temporal (loss of synchrony) as well as spatial [40]. Progress on the 2D transformation techniques will be especially relevant for analyzing these phenomena.

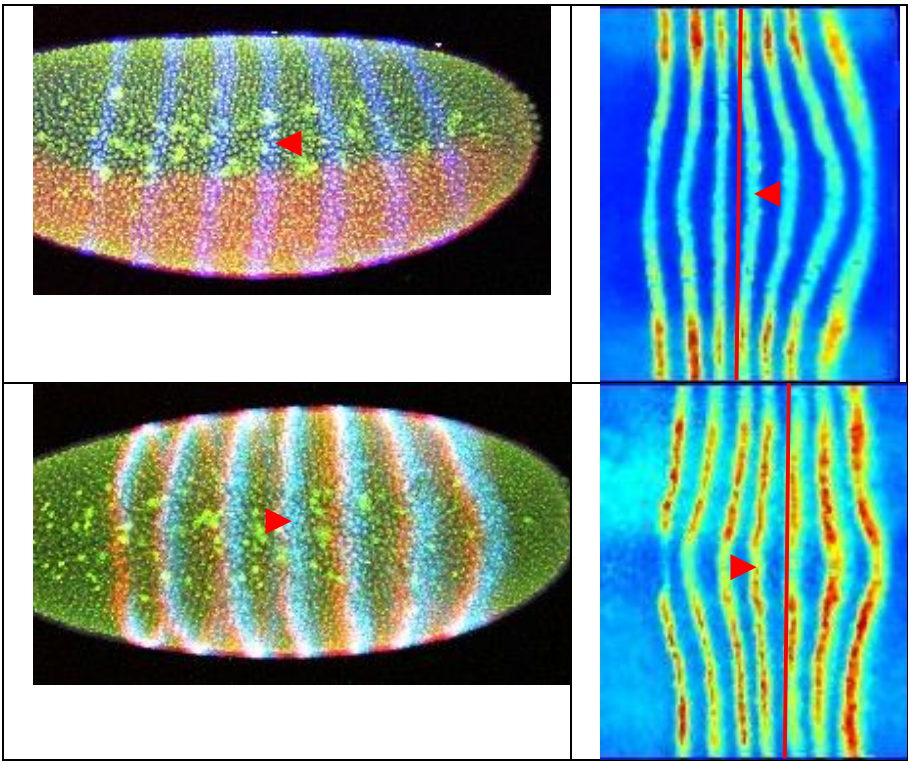


Fig. 2. Variability of intrinsic biological coordinates, as seen in *eve* patterns from two embryos. The fourth stripe (red arrows) can be curved to the left (head end of embryo) or to the right (tail end), red lines are drawn to show the stripe's curvature (BID BDTNP [29] embryos). I.e., the straight stripe forming the y-axis of the coordinate transformations can vary – here we see it at the 3rd stripe in one embryo and the 5th stripe in another.

Finally, the approach described with respect to Fig. 1, to transform between FlyEx and BID BDTNP types of data, allows for a much richer combined dataset: FlyEx contains chiefly protein data, while BID contains mRNA data. And while BDTNP has intact embryos, best for studying geometric effects, the flattened embryos have more accurate and sensitive detection of signal. The two approaches are complementary for many problems, and coordinate transformation between them can be an important tool for such investigations.

5 Challenges and New Developments

Our rotation & elastic deformation approach to 2D data: We are extending the approaches described above to use elastic deformation and rotation to fit 2D data of one embryo to another (flattened embryo data or cylindrical projections of intact embryos). A superposition of one embryo surface to another has several challenges.

Embryos differ in: spatial dimension (either in physical, micrometer, units, or in biological ones of nuclei numbers); nuclear density or total amount of nuclei; and in pattern features (a small but biologically significant factor). The procedure should be able to match embryos by patterns alone, or by patterns and nuclear positions together. There should also be freedom in choosing the spatial coordinates along which to optimize matching. Three operations should be able to match an embryo pair: horizontal and vertical shifting; rotation; and elastic deformation. These appear simple enough, but the high variability of embryo geometry and expression patterns makes the optimization tasks very hard. Some proportion of pairs will be very similar and matching gene patterns will give nearly perfect matches of nuclear positions. The larger proportion of pairs, however, even for coincident patterns, will not have coincident nuclei. This indicates deeper biological questions regarding the correlation between cell order and expression patterns, in addition to being a challenge to data processing.

6 Conclusions

Drosophila confocal image banks are not the only resources to which the approaches described here could be applied. Similar datasets exist for confocal scans of gene expression in other model organisms [1, 10]. We hope that the transformation techniques discussed here can also be applied to such cases.

Quantitative models of gene regulation are an integral part of understanding the mechanisms underlying functional genomics. *Drosophila* currently offers the highest resolution quantitative data available for validating models. This allows models to be tested on: the reduction of molecular noise during gene expression; the effects of cell movements and cell order on the developmental program; and the natural limits of reproducibility for gene expression patterns between embryos (as well as the effects of mutation on these limits). All of these efforts require the highest degree of quality from complex data sets. The techniques presented here have been developed to solve specific problems in the standardization and analysis of the biological data, so that such theoretical approaches can be tested, deepening the understanding of how genomes function in the development of tissues, organs, and individuals.

Acknowledgements. This work was supported by Joint NSF/NIGMS BioMath Program, 1-R01-GM072022 and the National Institutes of Health, 2R56GM072022-06, 2-R01-GM072022, CRDF - RUB1-33054-ST-12.

References

1. De Boer, B.A., Ruijter, J.M., Voorbraak, F.P.J.M., Moorman, A.F.: More than a decade of developmental gene expression atlases: where are we now? Nucleic Acids Res. 37, 7349–7359 (2009)
2. Fowlkes, C.C., LuengoHendriks, C.L., Keränen, S.V.E., Weber, G.H., et al.: A Quantitative Spatio-temporal Atlas of Gene Expression in the Drosophila Blastoderm. Cell 133, 364–374 (2008)

3. Pereanu, W., Hartenstein, V.: Digital three-dimensional models of Drosophila development. Curr. Opin. Genet. Dev. 14, 382–391 (2004)
4. Pawley, J.B. (ed.): Handbook of Biological Confocal Microscopy, 3rd edn. Springer, Berlin (2006)
5. Spirov, A.V., Kazansky, A.B., Timakin, D.L., Reinitz, J., Kosman, D.: Reconstruction of the dynamics of the Drosophila genes from sets of images sharing a common pattern. Special Issue on Imaging In Bioinformatics. Journal of Real-Time Imaging 8, 507–518 (2002)
6. Spirov, A.V., Holloway, D.: Evolutionary techniques for image processing to construct integrated dataset of early Drosophila embryo genes activity. EURASIP Journal on Applied Signal Processing 8, 824–833 (2003)
7. Surkova, S., Kosman, D., Kozlov, K., Manu, M.E., Samsonova, A.A., Spirov, A., Vanario-Alonso, C.E., Samsonova, M., Reinitz, J.: Characterization of the Drosophila segment determination morphome. Dev. Biol. 313, 844–862 (2008)
8. Kozlov, K., Myasnikova, E., Pisarev, A., Samsonova, M., Reinitz, J.: A method for two-dimensional registration and construction of the two-dimensional atlas of gene expression patterns in situ. Silico Biol. 2, 125–141 (2002)
9. Sorzano, C.O.S., Blagov, M., Thevenaz, P., Myasnikova, E., Samsonova, M., Unser, M.: Algorithm for Spline-Based Elastic Registration in Application to Confocal Images of Gene Expression. Pattern Recognition and Image Analysis 16, 93–96 (2006)
10. Preibisch, S.W., Saalfeld, S., Schindelin, J., Tomancak, P.: Software for bead-based registration of selective plane illumination microscopy data. Nat. Methods 7, 418–419 (2010)
11. Spirov, A.V., Timakin, D.L., Reinitz, J., Kosman, D.: Using of Evolutionary Computations in Image Processing for Quantitative Atlas of Drosophila Genes Expression. In: Proceedings of Third European Workshop on Evolutionary Computation In Image Analysis and Signal Processing, Milan, pp. 374–383 (2001)
12. Spirov, A.V., Timakin, D.L., Reinitz, J., Kosman, D.: Experimental Determination of Drosophila Embryonic Coordinates by Genetic Algorithms, the Simplex Method, and Their Hybrid. In: Oates, M.J., Lanzi, P.L., Li, Y., Cagnoni, S., Corne, D.W., Fogarty, T.C., Poli, R., Smith, G.D. (eds.) EvoIASP 2000, EvoWorkshops 2000, EvoFlight 2000, EvoSCONDI 2000, EvoSTIM 2000, EvoTEL 2000, and EvoROB/EvoRobot 2000. LNCS, vol. 1803, pp. 97–106. Springer, Heidelberg (2000)
13. Thompson, D.W.: On Growth and Form. Dover reprint of 2nd edn., 1942, 1st edn., 1917) (1992)
14. Holloway, D.M., Harrison, L.G., Kosman, D., Vanario-Alonso, C.E., Spirov, A.V.: Analysis of pattern precision shows that Drosophila segmentation develops substantial independence from gradients of maternal gene products. Dev. Dyn. 235, 2949–2960 (2006)
15. Zamparo, L., Perkins, T.J.: Statistical lower bounds on protein copy number from fluorescence expression images. Bioinformatics 25, 2670–2676 (2009)
16. Myasnikova, E., Samsonova, M., Kosman, D., Reinitz, J.: Removal of background signal from in situ data on the expression of segmentation genes in Drosophila. Dev. Genes Evol. 215, 320–326 (2005)
17. Surkova, S., Myasnikova, E., Janssens, H., et al.: Pipeline for acquisition of quantitative data on segmentation gene expression from confocal images. Fly 2, 58–66 (2008)
18. Poustelnikova, E., Pisarev, A., Blagov, M., Samsonova, M., Reinitz, J.: A database for management of gene expression data in situ. Bioinformatics 20, 2212–2221 (2004)
19. [ ]Lecuyer, E., Yoshida, H., Parthasarathy, N., Alm, C.et al: Global analysis of mRNA localization reveals a prominent role in organizing cellular architecture and function. Cell 131 (2007) 174-187

20. Zinzen, R.P., Senger, K., Levine, M., Papatsenko, D.: Computational models for neurogenic gene expression in the Drosophila embryo. Current Biology 16, 1358–1365 (2006)
21. Gregor, T., Bialek, W., de Ruyter van Steveninck, R.R., Tank, D.W., Wieschaus, E.F.: Diffusion and scaling during early embryonic pattern formation. Proc. Natl. Acad. Sci (USA) 102, 18403–18407 (2005)
22. He, F., Wen, Y., Deng, J., Lin, X., Lu, L.J., Jiao, R., Ma, J.: Probing intrinsic properties of a robust morphogen gradient in Drosophila. Dev. Cell 15, 558–567 (2008)
23. Crauk, O., Dostatni, N.: Bicoid determines sharp and precise target gene expression in the Drosophila embryo. Current Biol. 15, 1888–1898 (2005)
24. Arvey, A., Hermann, A., Hsia, C.C., Ie, E., Freund, Y., McGinnis, W.: Minimizing off-target signals in RNA fluorescent in situ hybridization. Nucleic Acids Res. (2010), doi:10.1093/nar/gkq042
25. Akam, M.: The molecular basis for metameric pattern in the Drosophila embryo. Development 101, 1–22 (1987)
26. Lawrence, P.A.: The making of a fly. London, 228 p. Blackwell Scientific (1992)
27. Panzer, S., Weigel, D., Beckendorf, S.K.: Organogenesis in Drosophila melanogaster: embryonic salivary gland determination is controlled by homeotic and dorsoventral patterning genes. Development 114, 49–57 (1992)
28. Skeath, J.B., Panganiban, G.F., Carroll, S.B.: The ventral nervous system defective gene controls proneural gene expression at two distinct steps during neuroblast formation in Drosophila. Development 120, 1517–1524 (1994)
29. `http://bdtnp.lbl.gov/Fly-Net/bioimaging.jsp`
30. `http://urchin.spbcas.ru/flyex/`
31. `http://fly-fish.ccbr.utoronto.ca`
32. Preibisch, S.W., Saalfeld, S., Rohlfing, T., Tomancák, P.: Bead-based mosaicing of single plane illumination microscopy images using geometric local descriptor matching. In: SPIE Medical Imaging Conference, Lake Buena Vista, Fla, pp. 1–10 (2009)
33. LuengoHendriks, C.L.: KeränenS.V.E., Fowlkes, C.C., Simirenko, L.et al.: Three-dimensional morphology and gene expression in the Drosophilablastoderm at cellular resolution I: data acquisition pipeline. Genome Biology 7, R123 (2006), doi:10.1186/gb-2006-7-12-r123
34. Bookstein, F.L.: Principal Warps: Thin-Plate Splines and Decomposition of Deformations. IEEE Trans. Pattern Analysis and Machine Intelligence 11, 567–585 (1989)
35. Siegel, A.F.: Geometric data analysis: An interactive graphics program for shape comparison. In: Launer, R.L., Siegel, A.F. (eds.) Modern Data Analysis, pp. 103–122. Academic Press (1981)
36. Siegel, A.F., Benson, R.H.: A robust comparison of biological shapes. Biometrics 38, 341–350 (1982)
37. Bookstein, F.L.: Morphometric Tools for Landmark Data: Geometry and Biology, p. 435. Cambridge U. Press, Cambridge (1991)
38. Abbasi, R., Mashhadihan, M., Abbasi, M., Kiabi, B.: Geometric morphometric study of populations of the social wasp, Polistesdominulus (Christ, 1791) from Zanjan province, north-west Iran. New Zealand Journal of Zoology 36, 41–46 (2009)
39. Keränen, S.V.E., Fowlkes, C.C., LuengoHendriks, C.L., Sudar, D., Knowles, D.W., Malik, J., Biggin, M.D.: Three-dimensional morphology and gene expression in the Drosophilablastoderm at cellular resolution II: dynamics. Genome Biology 7, R124 (2006), doi:10.1186/gb-2006-7-12-r124
40. Zallen, J.A., Zallen, R.: Cell-pattern disordering during convergent extension in Drosophila. Journal of Physics: Condensed Matter 16, S5073–S5080 (2004)

Identifying Informative Genes for Prediction of Breast Cancer Subtypes

Iman Rezaeian[1], Yifeng Li[1], Martin Crozier[2], Eran Andrechek[3], Alioune Ngom[1], Luis Rueda[1], and Lisa Porter[3]

[1] School of Computer Science, University of Windsor,
401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{rezaeia,li11112c,lrueda,angom}@uwindsor.ca
[2] Department of Biological Sciences, University of Windsor,
401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{mcrozier,lporter}@uwindsor.ca
[3] Department of Physiology, Michigan State University,
567 Wilson Rd, East Lansing, MI, 48824, United States
andrech1@msu.edu

Abstract. It is known that breast cancer is not just one disease, but rather a collection of many different diseases occurring in one site that can be distinguished based in part on characteristic gene expression signatures. Appropriate diagnosis of the specific subtypes of this disease is critical for ensuring the best possible patient response to therapy. Currently, therapeutic direction is determined based on the expression of characteristic receptors; while cost effective, this method is not robust and is limited to predicting a small number of subtypes reliably. Using the original 5 subtypes of breast cancer we hypothesized that machine learning techniques would offer many benefits for feature selection. Unlike existing gene selection approaches, we propose a tree-based approach that conducts gene selection and builds the classifier simultaneously. We conducted computational experiments to select the minimal number of genes that would reliably predict a given subtype. Our results support that this modified approach to gene selection yields a small subset of genes that can predict subtypes with greater than 95% overall accuracy. In addition to providing a valuable list of targets for diagnostic purposes, the gene ontologies of selected genes suggest that these methods have isolated a number of potential genes involved in breast cancer biology, etiology and potentially novel therapeutics.

Keywords: breast tumor subtype, gene selection, classification.

1 Introduction

Despite advances in treatment, breast cancer remains the second leading cause of cancer related deaths among females in Canada and the United States. Previous studies have revealed that breast cancer can be categorized into at least five subtypes, including basal-like (Basal), luminal A, (LumA), luminal B (LumB), HER2-enriched (HER2), and normal-like (Normal) types [1, 2]. These subtypes have their own genetic signatures, and response to therapy varies dramatically from one subtype to another. The

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 138–148, 2013.

variability among subtypes holds the answer to how to better design and implement new therapeutic approaches that work effectively for all patients. It is clinically essential to move toward effectively stratifying patients into their relevant disease subtype prior to treatment.

Techniques such as breast MRI, mammography, and CT scan, can examine the phenotypical mammary change, but provide little effective information to direct therapy. Genomic techniques provide high-throughput tools in breast cancer diagnosis and treatment, allowing clinicians to investigate breast tumors at a molecular level. The advance of microarray approaches have enabled genome-wide sampling of gene expression values and/or copy number variations. The huge amount of data that has been generated has allowed researchers to use unsupervised machine learning approaches to discover characteristic "signatures" that have since established distinct tumor subtypes [1]. Tumor subtyping has explained a great deal about some of the mysteries of tumor pathology [3], and has begun to enable more accurate predictions with regard to response to treatment [4]. While offering enormous opportunity for directing therapy, there are some challenges arising in the analysis of microarray data. First, the number of available samples (e.g. patients) is relatively small compared to the number of genes measured. The sample size typically ranges from tens to hundreds because of costs of clinical tests or ethical constraints. Second, microarray data is noisy. Although the level of technical noise is debatable [5], it must be carefully considered during any analysis. Third, due to technical reasons, the data set may contain missing values or have a large amount of redundant information. These challenges affect the design and results of microarray data analysis.

This current study focuses on identifying a minimal number of genes that will reliably predict each of the breast cancer subtypes. Being a field of machine learning, pattern recognition can be formulated as a feature selection and classification problem for multi-class, high-dimensional data using two traditional schemes. The first applies a multi-class "feature selection" method directly followed by a classifier to measure the dependency between a particular feature and the multi-class information. A well-known example of the feature selection method is the minimum redundancy maximum relevance (mRMR) method proposed in [6] and [7]. The second traditional scheme is the most common of the two and treats the multi-class feature selection as multiple binary-class selections. Methods using multiple binary class selections differ in how to bisect the multiple classes. The two most popular ways to solve this problem are one-versus-one and one-versus-all [8]. In this paper, we propose a novel and flexible hierarchial framework to select discriminative genes and predict breast tumor subtypes simultaneously. The main contributions of this paper can be summarized as follows:

1. We implement our framework using *Chi2* feature selection [9] and a *support vector machine (SVM) classifier* [10] to obtain biologically meaningful genes, and to increase the accuracy for predicting breast tumor subtypes.
2. We Use a novel feature selection scheme with a hierarchial structure, which learns in a cross-validation framework from the training data.
3. We establish a flexible model where any feature selection and classifier can be embedded for use.

4. We discover a new, compact set of biomarkers or genes useful for distinguishing among breast cancer types.

2 Related Work

Using microarray techniques, scientists are able to measure the expression levels for thousands of genes simultaneously. Finding relevant genes corresponding to each type of cancer is not a trivial task. Using hierarchical clustering, Perou and colleagues developed the original 5 subtypes of breast cancer based on the relative expression of 500 differentially expressed genes [1]. It has since been demonstrated that combining platforms to include DNA copy number arrays, DNA methylation, exome sequencing, microRNA sequencing and reverse-phase protein arrays may define these subtypes even further [2]. It is postulated that there are, indeed, upward of over 10 different forms of breast cancer with differing prognosis [25]. Other groups have tailored analysis toward refining the patient groups based on relative prognosis, reducing the profile for one subtype to a 14-gene signature [26]. Given any patient subtype, obtained through one or several platforms, we hypothesize that machine learning approaches can be used to more accurately determine the number of genes required to reliably predict a subtype for a given patients.

On the other hand, modeling today's complex biological systems requires efficient computational techniques designed in articulated model, and used to extract valuable information from existing data. In this regard, pattern recognition techniques in machine learning provide a wealth of algorithms for feature extraction and selection, classification and clustering. A few relevant approaches are briefly discussed then.

An entropy-based method for classifying cancer types was proposed in [16]. In entropy-classed signatures, the genes related to the different cancer subtypes are selected, while the redundancy between genes is reduced simultaneously. Recursive feature addition (RFA) has been proposed in [17], which combines supervised learning and statistical similarity measures to select relevant genes to the cancer type. A mixture classification model containing a two-layer structure named as mixture of rough set (MRS) and support vector machine (SVM) was proposed in [18]. This model is constructed by combining rough sets and SVM methods, in such a way that the rough set classifier acts as the first layer to determine some singular samples in the data, while the SVM classifier acts as the second layer to classify the remaining samples. In [19], a binary particle swarm optimization (BPSO) was proposed. BPSO involves a simulation of the social behavior in organisms such as bird flocking and fish schooling. In BPSO, a small subset of informative genes is selected where the genes in the subset are relevant for cancer classification. In [20], a method for selecting relevant genes in comparative gene expression studies was proposed, referred to as *recursive cluster elimination* (RCE). RCE combines k-Means and SVM to identify and score (or rank) those gene clusters for the purpose of classification. k-Means is used initially to group the genes into clusters. RCE is then applied to iteratively remove those clusters of genes that contribute the least to classification accuracy. In the work described in this paper we used the original five breast cancer subtypes to determine whether our proposed hierarchial tree-based scheme could reduce the gene signature to a reliable subset of relevant genes.

3 Methods

First, we describe the training phase for gene selection and breast cancer subtyping, and then we describe how the model can be used in predicting subytpes in a clinical setting. The complete gene profile of each breast cancer subtype is compared against the others. Each subtype varies in the genes that are associated with it, and in the accuracy with which those genes predict that specific subtype. The subtypes are then organized by two main criteria. The first criterion is the level of accuracy with which the selected genes identify the given subtype. The second criterion is the number of genes identified. Clearly applying two or more gene selection criteria is a multi-objective problem in optimization [21]. In this study, we use the rule that select the smallest subset of genes that yields the highest accuracy. Therefore, a subtype that is predicted with 95% accuracy by five genes is ranked higher than a subtype for which 20 genes are required to acquire the same accuracy. The subtype that is ranked highest is removed and the procedure is repeated for the remaining subtypes comparing each gene profile against the others. The highest ranked subtype is again removed and becomes a leaf on the hierarchical tree (see Fig. 1). Therefore, each leaf on the tree becomes a distinct subtype outcome.

3.1 Training Phase

We give an example of such a tree to illustrate our method in Fig. 1. Suppose there are five subtypes, namely $\{C_1, \cdots, C_5\}$. The training data is a $m \times n$ matrix $\boldsymbol{D} = \{\boldsymbol{D}_1, \cdots, \boldsymbol{D}_5\}$ corresponding to the five subtypes. $\boldsymbol{D}_i$, of size $m \times n_i$, is the training data for class C_i. m is the number genes and n_i is the number of samples in subtype C_i. $n = \sum_{i=1}^{5} n_i$ is the total number of training samples from all five classes. First of all, feature selection and classification are conducted, in a cross-validation fashion, for each class against the other classes. For example, suppose subtype C_3 obtains the highest rank based on accuracy and the number of genes contributing to that accuracy. We thus record the list of the particular genes selected and create a leaf for that subtype. We then remove the samples of the subtype, which results in $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_4, \boldsymbol{D}_5\}$ and continue the process in the same fashion. Thus, at the second level, subtype C_5 yields the highest rank, and hence its gene list is retained and a leaf is created. Afterward the training data set becomes $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_4\}$ for the third level. We repeat the training procedure in the same fashion until there is no subtype to classify. At the last level, two leaves are created, for C_4 and C_2, respectively.

3.2 Prediction Phase

Once the training is complete, we can apply the scheme to predict breast cancer subtypes. Given the gene expression profile of a new patient, a sequence of classification steps are performed by tracing a path from the root of the tree toward a leaf. At each node in the path, only the genes selected in the training phase are tested. The process starts at the first level (root of the tree), in which case only the genes selected for C_3, namely G_3 are tested. If the patient's gene profile is classified as a positive sample, then the prediction outcome is subtype C_3, and the prediction phase terminates. Otherwise, the sequence of classification tests is performed in the same fashion, until a leaf

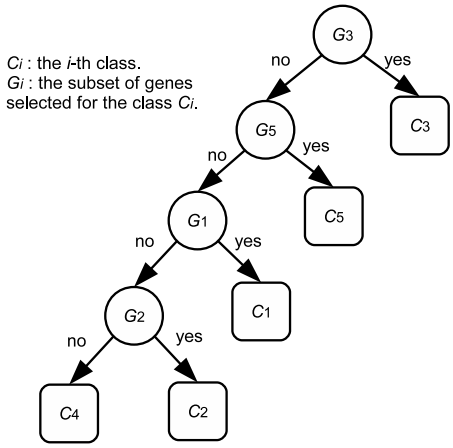


Fig. 1. Determining breast cancer type using selected genes

is reached, in which case the prediction outcome is the subtype associated with the leaf that has been reached.

3.3 Characteristics of the Method

Our structured model has the following characteristics. First, it involves a greedy scheme that tries the subtype which obtains the most reliable prediction and the smallest number of genes first. Second, it conducts feature selection and classification simultaneously. Essentially, it is a specific type of decision tree for classification. The differences between the proposed model and the traditional decision tree includes: i) each leaf is unique, while one class usually has multiple leaves in the later; ii) classifiers are learned at each node, while the traditional scheme learns decision rules; and iii) multiple features can be selected, while in the traditional scheme each node corresponds to only one feature. Third, the proposed model is flexible as any feature selection method and classifier can be embedded. Obviously, a classifier that can select features simultaneously also applies, (e.g. the l_1-norm SVM [11]).

3.4 Implementation

In this study, we implement our model by using Chi2 feature selection [9] and the state-of-the-art SVM classifier [10]. These two techniques are briefly described briefly next. Chi2 is an efficient feature selection method for numeric data. Unlike some traditional methods which discretize numeric data before conducting feature selection, Chi2 *automatically* and *adaptively* discretizes numeric features and selects features as well. It keeps merging adjacent discrete statuses with the lowest χ^2 value until all χ^2 values exceed their confidence intervals determined by a decreasing significant level, while keeping consistency with the original data. If, finally, a feature has only one discrete

status, it is removed. The χ^2 value of a pair of adjacent discrete statuses or intervals is computed by the χ^2 statistic, with 1 degree of freedom, as follows:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(n_{ij} - e_{ij})^2}{e_{ij}}, \tag{1}$$

where n_{ij} is the number of samples in the i-th interval and j-th class, and e_{ij} is the expected value of n_{ij}. e_{ij} is defined as $r_i \frac{c_j}{n}$ where $r_i = \sum_{j=1}^{k} n_{ij}$, $c_j = \sum_{i=1}^{2} n_{ij}$, and n is the total number training samples.

Based on these selected genes, the samples are classified using SVM [10]. Soft-margin SVM is applied in our current study. SVM is a linear maximum-margin model with decision function $d(\boldsymbol{x}) = \text{sign}[f(\boldsymbol{x})] = \text{sign}[\boldsymbol{w}^{\text{T}}\boldsymbol{x} + b]$ where $\boldsymbol{w}$ is the normal vector of the separating hyperplane and b is the bias. Soft-margin SVM solves the following problem in order to obtain the optimal $\boldsymbol{w}$ and b:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \boldsymbol{C}^{\text{T}}\boldsymbol{\xi} \\ \text{s.t. } & \boldsymbol{Z}^{\text{T}}\boldsymbol{w} + b\boldsymbol{y} \geq \boldsymbol{1} - \boldsymbol{\xi} \\ & \boldsymbol{\xi} \geq 0, \end{aligned} \tag{2}$$

where $\boldsymbol{\xi}$ is a vector of slack variables, $\boldsymbol{C}$ is a vector of constant that controls the trade-off between the maximum margin and the empirical error, $\boldsymbol{y}$ is a vector that contains the class information (either -1 or +1), and $\boldsymbol{Z}$ contains the normalized training samples with its i-th column defined as $\boldsymbol{z}_i = y_i\boldsymbol{x}_i$ [13]. Since optimization of the SVM involves inner products of training samples, by replacing the inner products by a kernel function, we can obtain a kernelized SVM.

For the implementation, the Weka machine learning suite was used [14]. A gene selection method based on the χ^2 feature evaluation algorithm was first used to find a subset of genes with the best ratio of accuracy/gene number [9]. For classification, LIBSVM [15] in Weka is employed. The *Radial basis function* (RBF) kernel is used with the LIBSVM classifier without normalizing samples and with default parameter settings.

4 Computational Experiments and Discussions

4.1 Experiments

In our computational experiment, we analyzed Hu's data [12]. Hu's data (CEO accession number GSE1992) were generated by three different platforms including Agilent-011521 Human 1A Microarray G4110A (feature number version) (GPL885), Agilent-012097 Human 1A Microarray (V2) G4110B (feature number version) (GPL887), and Agilent Human 1A Oligo UNC custom Microarrays (GPL1390). Each platform contains 22,575 probesets, and there are 14,460 common probesets among these three platforms. We used SOURCE [22] to obtain 13,582 genes with unique unigene IDs in order to merge data from different platforms. The dataset contains 158

samples from five subtypes of breast cancer(13 Normal, 39 Basal, 22 Her2, 53 LumA and 31 LumB). The sixth subtype Claudin is excluded from our current analysis as the number of samples of this class is too few (only five). However, we will investigate this subtype in our future work.

To evaluate the accuracy of the model, 10-fold cross-validation is used. As shown in Table 2, using all genes decreases the overall accuracy of the model, since many of the genes are irrelevant or redundant. For example, using all 13,582 genes, the overall accuracy is just 77.84%; while using a ranking algorithm and taking the top 20 genes for prediction brings the accuracy up to 86.70%. Table 1 shows the top 20 genes ranked by the Chi-Squared attribute evaluation algorithm to classify samples as one of the five subtypes. Using the proposed hierarchical decision-tree-based model, makes the prediction procedure more accurate. While the accuracy of prediction between LumA and LumB is relatively low compared to the other classes. This is because of the very high similarity and overlap between samples of these two classes. The overall accuracy of the model, as shown in Table 2, is 95.11%. This is very interesting since only 18 genes are used to predict the subtypes that the patient belongs to. As a matter of fact, our method is able to increase its accuracy from around 86% to 95% by using a new subset of genes based on the proposed method containing only 18 genes.

Table 1. Top 20 genes ranked by the Chi-Squared attribute evaluation algorithm to classify samples as one of the five subtypes

Rank	Gene Name	Rank	Gene Name	Rank	Gene Name	Rank	Gene Name
1	FOXA1	6	THSD4	11	DACH1	16	ACOT4
2	AGR3	7	NDC80	12	GATA3	17	B3GNT5
3	CENPF	8	TFF3	13	INPP4B	18	IL6ST
4	CIRBP	9	ASPM	14	TTLL4	19	FAM171A1
5	TBC1D9	10	FAM174A	15	VAV3	20	CYB5D2

Fig. 2 shows the tree learned in the training phase and the set of genes selected at each step. The selected genes are contained in each node, a patient's gene expression profile is used to feed the tree for prediction, each leaf represents a subtype, and the accuracy at each classification step is under the corresponding node. From this figure, we can see that the Basal subtype is chosen first as it obtains the highest accuracy, 99.36% to classify patients from the other subtypes including Normal, Her2, LumA and LumB. Then the samples of Basal are removed for the second level. The Normal subtype is chosen then, since it achieves the highest accuracy (95.79%) to separate samples from the other subtypes, including Her2, LumA and LumB. From previous studies, it is well-known that the subtypes LumA and LumB are very difficult to be identified among all subtypes. This is the reason for why LumA and LumB appear at the bottom of the tree. After removing other subtypes, LumA and LumB can avoid misclassification on the other subtypes. In spite of this drawback, the accuracy for separating LumA and LumB is as high as 88.1%.

As shown in Figure 2, there is no overlap between the genes selected among the different clusters. This result provides interesting new biomarkers for each breast cancer

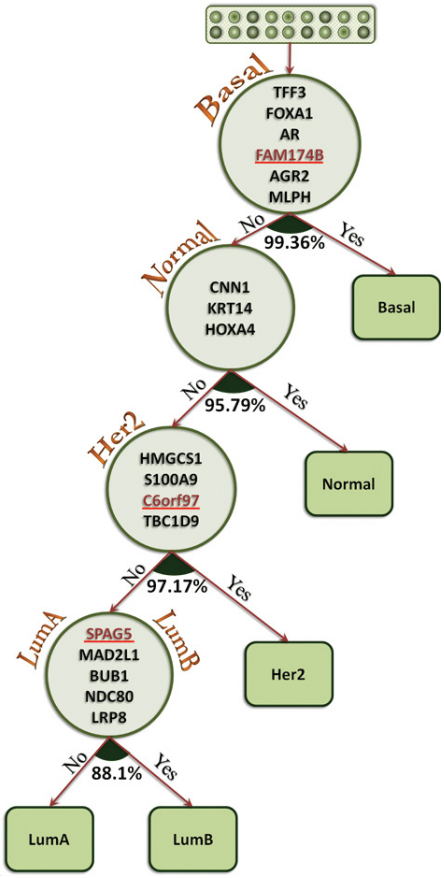


Fig. 2. Determining breast cancer type using selected genes

subtype. Some of the selected genes have been previously indicated in cancer (highlighted in black in Figure 2), while others have emerged as interesting genes to be investigated. For example, TFF3 and FoxA1 genes are predictably indicated in Basal subtype. Another feature of the proposed hierarchical model is that the number of genes in each node has been optimized to give the best ratio of accuracy and number of selected genes. For this, at first, 10 genes with highest rank have been selected for each node. Then, out of those selected genes, those with lower rank are removed step by step as long as the accuracy of classification using the remaining genes don't get decreased.

4.2 Biological Insight

We used FABLE to determine if the genes selected by our approach are biologically meaningful. Fast Automated Biomedical Literature Extraction (FABLE) is a web-based tool to search through MEDLINE and PubMed databases. The genes that are related

Table 2. Accuracy of classification using LibSVM Classifier

Classification Method	Gene Selection Method	# of Genes	Accuracy	Precision	Recall	F-measure
LibSVM	—	all genes	77.84%	0.802	0.778	0.749
LibSVM	Chi-Squared	20	86.70%	0.866	0.867	0.864
Proposed Method	Proposed Method	18	95.11%	0.951	0.951	0.951

to tumors reported in the literature are highlighted in black in Figure 2. Those not yet reported are underlined and colored in red. We can see that 15 out of 18 genes have been found in the literature. This implies that our approach is quite effective in discovering new biomarkers.

We also explored the reasons for the high performance of our method. First, the subtypes that are easily classified are on the top of the tree, while the harder subtypes are considered only after removing the easier ones. Such a hierarchical structure can remove the disturbance of other subtypes, thereby allowing us to focus on the most difficult subtypes, LumA/B. Second, combining gene selection when building the classifier allows us to select genes that contribute to prediction accuracy. Third, our tree-based methodology is quite flexible; any existing gene selection measure and classification technique can be embedded in our model. This will allow us to apply this model to subtypes as they become more rigorously defined using other platforms such as copy number variation. Furthermore, our method could be applied to groups of patients stratified based on responses to specific treatments. Collectively, having a small, yet reliable number of genes to screen is more cost effective and would allow for subtype information to be more readily applied in a clinical setting.

5 Conclusion and Future Work

In this study, we proposed a novel gene selection method for breast cancer subtype prediction based on a hierarchical, tree-based model. The results demonstrate an impressive accuracy to predict breast cancer types using only 18 genes. Herein, we propose a novel gene selection method for breast cancer subtype prediction based on a hierarchical, tree-based model. The results demonstrate an impressive accuracy to predict breast cancer subtypes using only 18 genes in total. Moreover, Most of the selected genes are shown to be related to breast cancer based on previous studies, while a few are yet to be investigated. As future work, we will validate these results using cell lines that fall within a known subtype. We will determine whether our predicted 18 gene array can accurately denote which subtype each of these cell lines falls under. This hierarchical, tree-based model can narrow down analysis to a relatively small subset of genes. Importantly, the method can be applied to more refined stratification of patients in the future, such as subtypes derived using a combination of platforms, or for groups of patients that have been subdivided based on response to therapy. Using this computational tool we can determine the smallest possible number of genes that need to be screened for accurately placing large populations of patients into specific subtypes of cancer or specified treatment groups. This could contribute to the development of improved screening tools, providing increased accuracy for a larger patient population than that achieved by

Oncotype DX, but allowing for a cost effective approach that could be widely applied to the patient population.

Acknowledgments. This research has been supported by grants from Seeds4Hope (WECCF), CBCRA (#02051), and Canadian NSERC Grants #RGPIN228117-2011 and #RGPIN261360-2009.

References

1. Perou, C.M., et al.: Golecular Portraits of Human Breast Tumours. Nature 406, 747–752 (2000)
2. Perou, C.M., et al.: Comprehensive Molecular Portraits of Human Breast Tumours. Nature 490, 61–70 (2012)
3. Chandriani, S., Frengen, E., Cowling, V.H., Pendergrass, S.A., Perou, C.M., Whitfield, M.L., Cole, M.D.: A Core MYC Gene Expression Signatures is Prominent in Basal-Like Breast Cancer but only Partially Overlaps the Core Serum Response. PLOS One 4(8), e6693 (2009)
4. van't Veer, L.J., et al.: Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer. Nature 415(6871), 530–536 (2002)
5. Klebanov, L., Yakovlev, A.: How High is The Level of Technical Noise in Microarray Data? Biology Direct. 2, 9 (2007)
6. Ding, C., Peng, H.: Munimun Redundancy Feature Selection from Microarray Gene Expression Data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
7. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
8. Li, T., Zhang, C., Ogihata, M.: A Comparative Study of Feature Selection and Multiclass Classification Methods for Tissue Classification Vased on Gene Expression. Bioinformatics 20(15), 2429–2437 (2004)
9. Liu, H., Setiono, R.: Chi2: Feature Selection and Discretization of Numeric Attributes. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 388–391. IEEE Press, New York (1995)
10. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
11. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-Norm Support Vector Machines. In: NIPS. MIT Press, Cambridge (2004)
12. Hu, Z., et al.: The Molecular Portraits of Breast Tumors are Conserved Across Microarray Platforms. BMC Genomics 7, 96 (2006)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience, New York (2006)
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
15. Chang, C.-C., Lin, C.-J.: LIBSVM: a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology 12, 27:1–27:27 (2011)
16. Liu, X., Krishnan, A., Mondry, A.: An Entropy-Based Gene Selection Method for Cancer Classification Using Microarray Data. BMC Bioinformatics 6, 76 (2005)
17. Liu, Q., Sung, A.H., Chen, Z., Liu, J., Huang, X., Deng, Y.: Feature Selection and Classification of MAQC-II Breast Cancer and Multiple Myeloma Microarray Gene Expression Data. PLoS One 4(12), e8250 (2009)

18. Zeng, T., Liu, J.: Mixture Classification Model Based on Clinical Markers for Breast Cancer Prognosis. Artificial Intelligence in Medicine 48, 129–137 (2010)
19. Mohamad, M.S., Omatu, S., Deris, S., Yoshioka, M.: Particle Swarm Optimization for Gene Selection in Classifying Cancer Classes. Artificial Life and Robotics 14(1), 16–19 (2009)
20. Yousef, M., Jung, S., Showe, L., Showe, M.: Recursive Cluster Elimination (RCE) for Classification and Feature Selection from Gene Expression Data. BMC Bioinformatics 8, 144 (2007)
21. Li, Y., Ngom, A., Rueda, L.: A Framework of Gene Subset Selection Using Multiobjective Evolutionary Algorithm. In: Shibuya, T., Kashima, H., Sese, J., Ahmad, S. (eds.) PRIB 2012. LNCS (LNBI), vol. 7632, pp. 38–48. Springer, Heidelberg (2012)
22. Diehn, M., et al.: SOURCE: a Unified Genomic Resource of Functional Annotations, Ontologies, and Gene Expression Data. Nucleic Acids Research 31(1), 219–223 (2003), `http://smd.stanford.edu/cgi-bin/source/sourceSearch`
23. Sorlie, T., et al.: Gene Expression Patterns of Breast Carcinomas Distinguish Tumor Subclasses with Clinical Implications. PANS 98(19), 10869–10874 (2001)
24. Sorlie, T., et al.: Repeated Observation of Breast Tumor Subtypes in Independent Gene Expression Data Sets. PANS 100(14), 8418–8423 (2003)
25. Curtis, C., et al.: The Genomic and Transcriptomic Architecture of 2,000 Breast Tumours Reveals Novel Subgroups. Nature 486(7403), 346–352 (2012)
26. Hallett, R.M., Dvorkin-Gheva, A., Bane, A., Hassell, J.A.: A Gene Signature for Predicting Outcome in Patients with Basal-Like Breast Cancer. Scientific Reports 2, 227 (2012)

Predicting Therapeutic Targets with Integration of Heterogeneous Data Sources

Yan-Fen Dai[1,2], Yin-Ying Wang[1,3], and Xing-Ming Zhao[4,⋆]

[1] Institute of Systems Biology, Shanghai University, Shanghai 200444, China
[2] Department of Mathematics, Shanghai University, Shanghai 200444, China
[3] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
[4] School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
xm_zhao@tongji.edu.cn

Abstract. Drug target is of great importance for designing new drugs and understanding the molecular mechanism of drug actions. In general, a drug may bind to multiple proteins, some of which are not related to disease-treatment or even lead to side effects. Therefore, it is necessary to discriminate the effect-mediating drug targets, i.e. therapeutic targets, from other proteins. Although a lot of computational approaches have been developed to predict drug targets and achieve partial success, few attention has been paid to predict therapeutic targets. In this work, we present a new framework to predict drug therapeutic targets based on the integration of heterogeneous data sources. In particular, we develop an ensemble classifier, PTEC (Predicting Therapeutic targets with Ensemble Classifier), that can effeciently integrate both drug and protein properties described from distinct perspectives, thereby improving prediction accuracy. The results on benchmark datasets demonstrate that our approach outperforms other popular approaches significantly, implying the effectiveness of our proposed approach. Furthermore, the results indicate that the integration of different data sources can not only improve the coverage of predicted targets but also the prediction precision. In other words, distinct data sources indeed complement with each other, and the integration of these heterogeneous data sources can improve the prediction accuracy.

1 Introduction

Drug target identification is one of the most important steps in drug development, and is the key to understand how the desirable therapeutic effects are accomplished when the proteins are targeted by drugs [1,2]. Unfortunately, the targets of a lot of drugs are incomplete or even unknown, which hampers the discovery of new drugs. Recently, a number of computational approaches have been proposed to predict drug targets. For example, assuming similar drugs bind

⋆ Corresponding author.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 149–158, 2013.

to similar pockets on the protein surfaces, molecular docking approaches have been widely used to identify those compounds that can bind to known target proteins by investigating the chemical similarity between candidate ligands with known drugs [3]. With the knowledge that drugs with similar therapeutic effects generally target same proteins, drug therapy information has been used to predict drug targets [4]. Observing that drugs with similar side effects tend to target common proteins, Campillos *et al* proposed a novel approach to predict drug targets based on side effect similarity [5]. Considering that protein function is determined by its component domains while ligands generally bind to proteins to exert their function [6], Wang *et al* proposed a novel statistical approach to predict drug targets based on the derived interactions between drugs and protein domains [7]. To further improve prediction accuracy, different kinds of data sources have been integrated to predict compound-protein interactions. For example, Yamanishi *et al* have combined chemical structure and genomic sequence information to predict drug-protein interactions [8], and they later further took into account the pharmacological information to improve prediction accuracy [9].

With the knowledge about drug-protein interactions becoming more comprehensive, the amount of compound-protein interactions deposited in public databases, e.g. DrugBank [10] and STITCH [11], increases accordingly. Most recently, it is found that actually 96% of approved drugs have known targets [12]. However, a large number of these drug-protein interactions are found to be either irrelevant to disease-treatment or related to side effects [13]. In general, a compound may bind to multiple proteins, among which some proteins are off-targets that may lead to severe undesirable adverse effects. That is, druggable proteins are not necessarily main effect-mediating targets, i.e. therapeutic targets, that play critical and preferably unsubstitutable roles when treating disease [14]. Therefore, it is necessary to identify those therapeutic targets, and discriminate them from therapeutically irrelevant or side effect related ones. The therapeutic targets can help design drugs with expected efficacy. Although experimental techniques, such as high-throughout screening with bioassays, can be used to detect drug-protein interactions, it is highly expensive and time-consuming to identify the effect-mediating targets from the large pool of proteins within the human genome. Despite the partial success achieved by above mentioned computational approaches, few attention has been paid to predict therapeutic targets in the bioinformatics community possibly due to the scarceness of therapeutic target information.

In this paper, we present a novel framework to predict the therapeutic targets for known drugs based on integration of heterogeneous data sources. To this end, we investigate various properties of both drugs and proteins, including chemical structure and therapy information for drugs while primary structure and functional annotations for proteins. In particular, we develop a novel approach to integrate these heterogeneous data for both drugs and proteins with an ensemble classifier, PTEC (Predicting Therapeutic targets with Ensemble Classifier). The integration of different data sources can not only improve prediction

coverage but also accuracy [15]. That is, distinct data sources can complement with each other so that better results are expected based on the integration of these heterogeneous data sources. The results on gold standard datasets demonstrate that our proposed method outperforms other popular approaches significantly, implying the effectiveness of our proposed approach.

The rest of this paper is organized as following. Section 2 presents the materials used in this work and our proposed methods; Section 3 presents the experimental results; Finally, conclusions are drawn in Section 4.

2 Materials and Methods

2.1 Data Sources

In this work, 406 therapeutic targets for known drugs were retrieved from [12], which were curated from the drug-protein interactions from the DrugBank database [10]. We also downloaded other human drug target proteins and drug therapy information from DrugBank database (version 3.0). The drug therapy information described as therapeutic categories in Anatomic Therapeutic Chemical (ATC) classification system was considered here. The chemical structure information for drugs was obtained from PubChem [16]. As a result, 708 drugs with both chemical structure and therapy information available were kept for further analysis, which leads to 1726 interactions between drugs and their corresponding therapeutic targets.

The amino acid sequences of human proteins were obtained from the Uniprot database [17]. The functional annotations for these proteins were extracted from the Gene Ontology (GO) database [18], where all three functional categories were considered, including cellular component, molecular function and biological process. The protein associated pathway information was retrieved from KEGG database [19]. Furthermore, the expression profiles of protein coding genes generated for 36 normal human tissues were obtained from [20].

2.2 Drug Similarity

With chemical structure and therapy information available for drugs, we can define the similarity between two drugs. The chemical similarity between a pair of drugs was calculated as the two-dimensional Tanimoto score based on their fingerprints with the help of Chemistry Development Kit (CDK) [21], which is defined as following.

$$C_s(d, d^{'}) = \frac{\sum_i (d_i \wedge d_i^{'})}{\sum_j (d_j \vee d_j^{'})} \tag{1}$$

where $C_s(d, d^{'})$ represents the similarity score of two drugs d and $d^{'}$, d_i is the ith bit in the fingerprint of drug d, and $\wedge$ and $\vee$ respectively denotes bitwise $'and'$ and $'or'$ operators.

In the Anatomic Therapeutic Chemical (ATC) classification system, each drug can be described in 5 hierarchical levels and is classified into different therapeutic

groups according to the organ it acts on and its chemical characteristics. In this work, the therapeutic similarity between two drugs was defined as their longest matched prefix between their corresponding ATC codes as described previously [4].

$$T(d, d^{'}) = \max_{(d_i, d_j^{'})} \frac{2 * \log(Pr(pre(d_i, d_j^{'})))}{\log(Pr(d_i)) + \log(Pr(d_j^{'}))} \tag{2}$$

where $T(d, d^{'})$ denotes the therapeutic similarity between drugs d and $d^{'}$, d_i denotes the ith ATC category for drug d considering each drug may be grouped into different categories, $pre(i, j)$ denotes the longest matched prefix between the ATC codes d_i and $d_j^{'}$, $Pr(d_i)$ denotes the probability of the ATC category d_i occurs in drugs, and $Pr(pre(d_i, d_j^{'}))$ denotes the probability of the common prefix between the two ATC categories d_i and $d_j^{'}$ occurs in drugs.

2.3 Protein Similarity

The most straightforward way to measure the similarity between two proteins is to compare their primary structure identity. In this work, the sequence similarity $S_s(p, p^{'})$ between two proteins $(p, p^{'})$ is defined as the normalized Smith-Waterman alignment score as described as following.

$$S_s(p, p^{'}) = \frac{SS(p, p^{'})}{\sqrt{SS(p, p)SS(p^{'}, p^{'})}} \tag{3}$$

where $SS(.,.)$ denotes the original Smith-Waterman alignment score [22].

The pathways associated with drug target proteins can tell the molecular context in which the proteins exert their function, and therefore help to understand the mechanism of actions of drugs. With pathway annotation for proteins available, the pathway similarity $S_p(p, p^{'})$ between two proteins can be defined as below.

$$S_p(p, p^{'}) = \frac{\left| S(p) \cap S(p^{'}) \right|}{|S(p) \cup S(p^{'})|} \tag{4}$$

where $S(p)$ and $S(p^{'})$ respectively denotes the set of pathways in which protein p and $p^{'}$ are located.

Furthermore, with the functional annotations extracted from GO database, the functional similarity $S_g(p, p^{'})$ between two proteins p and $p^{'}$ is defined as the Jaccard index.

$$S_g(p, p^{'}) = \frac{\sum_{k=1}^{3} \left| t_k(p) \cap t_k(p^{'}) \right|}{\sum_{k=1}^{3} |t_k(p) \cup t_k(p^{'})|} \tag{5}$$

where $t_k(p)$ is the set of GO terms associated with protein p with respect to functional category k, $k = 1, 2, 3$ denotes each of the three functional categories in GO database, i.e. Molecular Function, Biological Process, and Cellular Component.

In addition, the expression similarity between two genes coding a pair of proteins was defined as coexpression correlation based on the gene expression profiles of 36 normal human tissues from [20] as below.

$$S_t(p, p^{'})) = \frac{\sum_{k=1}^{n}(p(k)-\overline{p})(p^{'}(k)-\overline{p^{'}})}{\sum_{k=1}^{n}\sqrt{(p(k)-\overline{p})^2(p^{'}(k)-\overline{p^{'}})^2}} \tag{6}$$

where $S_t(p, p^{'})$ is the correlation coefficient between the genes coding proteins p and $p^{'}$, n is the number of samples, and $\overline{p}$ is the mean of expression profile of protein p.

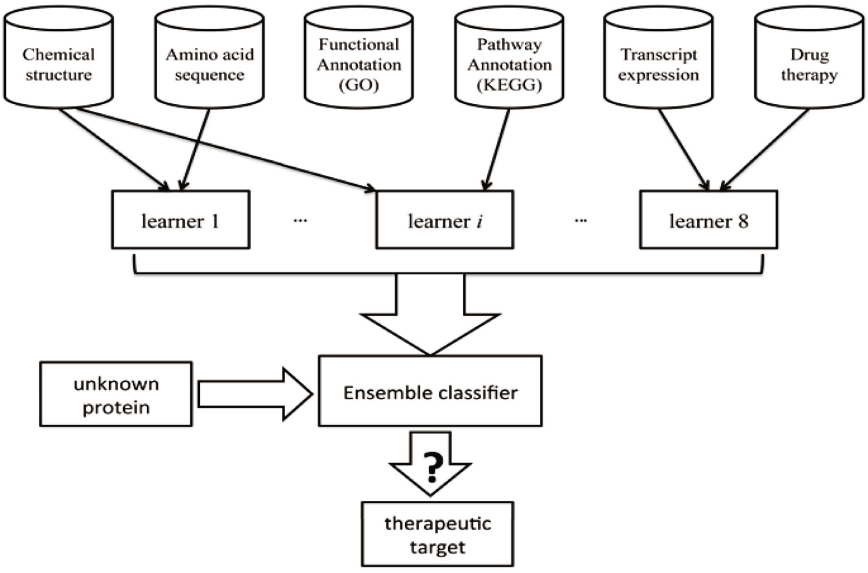


Fig. 1. The flowchart of predicting therapeutic targets based on the integration of heterogeneous data sources

2.4 Therapeutic Target Prediction

With the drug similarity described above, we assume that drugs with similar characteristics will target same proteins. Similarly, the proteins with similar properties will be bound by same drugs. With this in mind, we can construct a learner based on known drug-protein interactions. In this work, a drug-protein pair (d_i, p_j) can be represented as a feature vector (Fd_i, Fp_j), where each element in Fd_i represents the similarity between drug d_i and all the drugs while each element in Fp_j represents the similarity between protein p_i and all the proteins. For example, for the combination of chemical structure and protein

sequence, the elements in Fd_i denotes the chemical similarity between drug d_i and the rest drugs while the elements in Fp_j denotes the sequence similarity between protein p_j and the other proteins. After the feature extraction step, a classifier will be subsequently trained for each combination of drug and protein properties, e.g. drug therapy and protein sequence. In this way, we can have 8 different combinations between distinct drug and protein properties, thereby leading to 8 classifiers. Instead of selecting the best-performing classifier from the eight ones, we proposed to construct an ensemble classifier, PTEC (Predicting Therapeutic targets with Ensemble Classifier), to integrate these distinct learners in a weighted way (see Fig 1). The ensemble classifier was adopted here since it has been found to outperform individual ones and is more robust [23]. In particular, we first evaluated each classifier on a benchmark dataset, and used their accuracy as their corresponding weights to construct the ensemble classifier as following.

$$Enc_{res} = \sum_{i=1}^{8} W_i \cdot L_i \tag{7}$$

where Enc_{res} is the predicted results by the ensemble classifier, W_i is the weight for learner ith, and L_i is the output of learner ith. Here the weight for each learner is set to the area under the curve (AUC) score of a reciever operating characteristic (ROC) curve it obtained on the training set. Therefore, for a given unknown protein, we can use the Ensemble classifier to predict whether it is a therapeutic target. The simple but effective k-nearest neighbor algorithm (k-NN) was used as the learner in this work.

3 Results and Discussion

With the known interactions between drugs and their corresponding therapeutic targets as positive set, we build a negative set consists of drug-protein interactions from DrugBank except those from the positive set for the drugs involved in the positive set. As a result, 1094 drug-protein interactions were obtained as negative set. Note that all the drug-protein interactions in the negative set are real interactions as reported in DrugBank.

To evaluate the predictive power of different classifiers, one fifth of the samples were used as the test set while the rest were used as the training set. Firstly, we evaluated the eight single classifiers based on the training set with 10-fold cross-validation. Table 1 summarizes the results obtained by distinct classifiers. From the results, we can see that these eight classifiers perform comparably well with no one single classifier performs always best. For example, the classifier trained with therapy information and gene expression achieves the highest true positive rate, while the one trained on protein sequence performs best with respect to false negative rate. With the AUC scores obtained by the eight classifiers on the training set as their corresponding weights, we integrated the eight classifiers into an ensemble classifier PTEC, which achieves the highest true positive rate and the best overall result with an AUC score of 0.71 (see Table 1). The ensemble classifier

Table 1. Performance of distinct classifiers, where the results were obtained with 10-fold cross-validation on the training set

	C_{cs}	C_{cp}	C_{ct}	C_{cg}	C_{As}	C_{Ap}	C_{At}	C_{Ag}	**PTEC**
TPR	0.77	0.76	0.80	0.76	0.77	0.79	0.80	0.79	**0.81**
TPR_*std*	0.02	0.01	0.02	0.02	0.03	0.02	0.01	0.01	0.01
FPR	0.37	0.41	0.46	0.36	**0.37**	0.43	0.46	0.43	0.39
FPR_*std*	0.02	0.01	0.03	0.02	0.02	0.01	0.02	0.02	0.01
AUC	0.70	0.66	0.68	0.70	0.68	0.67	0.70	0.70	**0.71**
AUC_*std*	0.02	0.01	0.02	0.02	0.01	0.01	0.02	0.01	0.01

C_{cs} - classifier trained on chemical structure and protein sequence; C_{cp} - classifier trained on chemical structure and protein pathway; C_{ct} - classifier trained on chemical structure and transcriptional expression; C_{cg} - classifier trained on chemical structure and protein GO annotation; C_{As} - classifier trained on drug therapy information and protein sequence; C_{Ap} - classifier trained on therapy information and protein pathway; C_{At} - classifier trained on therapy information and transcriptional expression; C_{Ag} - classifier trained on therapy information and protein GO annotation;
TPR - true positive rate;
TPR_*std* - standard deviation of true positive rate;
FPR - false positive rate;
FPR_*std* - standard deviation of false positive rate;
AUC - Area under ROC curve;
AUC_*std* - standard deviation of AUC.

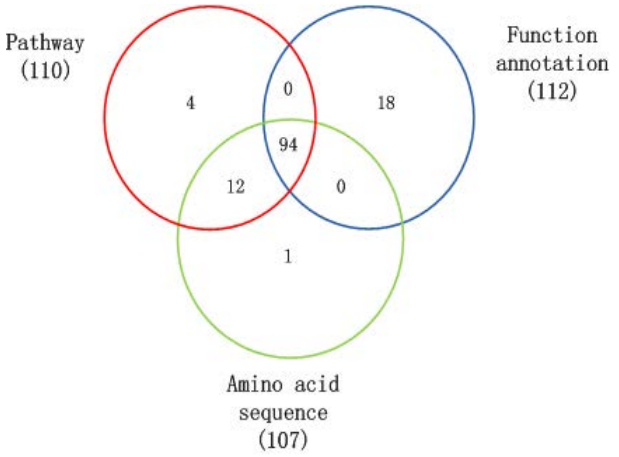


Fig. 2. The Venn diagram about the number of drug-protein interactions successfully predicted by the combination between drug therapy and three protein properties

was adopted here since it can improve prediction coverage considering that the annotations for proteins are incomplete. For example, looking into the drug-protein interactions predicted by different classifiers, Fig. 2 shows the Venn diagram about the number of drug-target pairs successfully predicted by the combination of drug therapy with protein sequence, pathway annotation and functional annotation respectively. It can be seen that among the 138 drug-protein interactions, the three

Table 2. Performance of distinct classifiers on the test set

	C_{cs}	C_{cp}	C_{ct}	C_{cg}	C_{As}	C_{Ap}	C_{At}	C_{Ag}	**PTEC**
TPR	0.70	0.78	0.79	0.80	0.76	0.80	0.82	0.82	**0.83**
FPR	0.26	0.26	0.36	0.29	0.32	0.33	0.38	0.34	**0.17**
AUC	0.76	0.75	0.71	0.75	0.73	0.74	0.72	0.74	**0.83**

classifiers get consistent results on most of their predictions 68.12% (94/138), while the integration of these different data sources can enlarge the number of predicted therapeutic targets significantly. In other words, distinct data sources complement with each other and the integration of them can improve both prediction accuracy and coverage.

To further evaluate the predictive power of our proposed PTEC, we applied it to predict therapeutic targets on the hold-out test set. Moreover, we compared our results with those eight single classifiers. Table 3 shows the performance of distinct classifiers on the test set. The results demonstrate that our proposed ensemble classifier significantly outperforms others with an AUC score of 0.83 and the highest true positive rate, indicating the effectiveness and robustness of our proposed ensemble classifier.

In addition, we compared our proposed method with a popular approach, namely nearest profile (NP), which predicts drug targets based on a bipartite graph. Figure 3 gives the results obtained by both PTEC and NP, where the

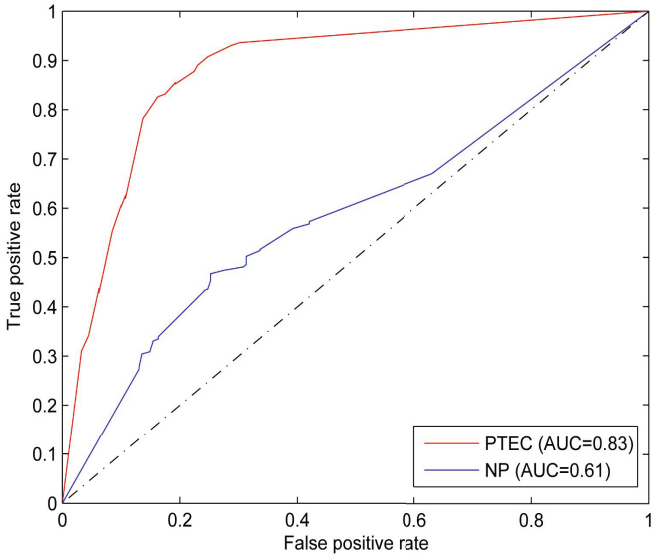


Fig. 3. The performance of PTEC and the nearest profile(NP) method

results by PTEC are based on the test set while those by NP are based on the whole dataset. From the results, we can clearly see that PTEC is really effective to predict therapeutic targets, and is able to separate therapeutic targets from other irrelevant ones. The good performance of PTEC confirm again that the integration of different data sources indeed can improve prediction accuracy and also the predictive power of our proposed approach.

In our predictions, some of them are not found in the positive dataset, which does not necessarily mean they are false positives. For example, we predict protein AchE that is involved in lipid transportation and metabolism as the therapeutic target of drug Physostol, a cholinesterase inhibitor that can be applied topically to the conjunctiva. In the positive set, AchE is not the therapeutic target of Physostol, whereas we found that AchE is reported as the therapeutic target of Physostol in the Therapeutic Target Database (TTD)[24]. The drug Metubine iodide is a benzylisoquinolinium competitive nondepolarizing neuromuscular blocking agent, which was predicted to bind to CHRNA2 by our proposed PTEC, and this interaction is also verified in TTD. The verification of our prediction results by other public databases demonstrates the predictive power of our proposed method.

4 Concluding Remarks

Therapeutic target is the key to design the drugs with expected efficiency and understand how the drugs work. In this paper, we present a new framework to predict drug therapeutic targets by integrating heterogeneous data sources for both drugs and proteins. Specifically, we proposed a novel ensemble classifier to integrate the learners trained on distinct data sources. The results on benchmark dataset demonstrate the effectiveness and robustness of our proposed approach.

Acknowledgement. This work was partly supported by the National Natural Science Foundation of China (91130032, 61103075), Innovation Program of Shanghai Municipal Education Commission (13ZZ072) and Innovation Program of Shanghai University(SHUCX120115).

References

1. Yildirim, M.A., Goh, K.I., Cusick, M.E., Barabasi, A.L., Vidal, M.: Drug-target network. Nat. Biotechnol. 25, 1119–1126 (2007)
2. Yabuuchi, H., Niijima, S., Takematsu, H., Ida, T., Hirokawa, T., Hara, T., et al.: Analysis of multiple compound-protein interactions reveals novel bioactive molecules. Mol. Syst. Biol. 7, 472 (2011)
3. Keiser, M.J., Setola, V., Irwin, J.J., Laggner, C., Abbas, A.I., et al.: Predicting new molecular targets for known drugs. Nature 462, 175–181 (2009)
4. Zhao, S., Li, S.: Network-based relating pharmacological and genomic spaces for drug target identification. PLoS ONE 5, e11764 (2010)

5. Campillos, M., Kuhn, M., Gavin, A.C., Jensen, L.J., Bork, P., et al.: Drug target identification using side-effect similarity. Science 321, 263–266 (2008)
6. Zhao, X.M., Chen, L., Aihara, K.: A discriminative approach for identifying domain-domain interactions from protein-protein interactions. Proteins 78, 1243–1253 (2010)
7. Wang, Y.Y., Nacher, J.C., Zhao, X.M.: Predicting drug targets based on protein domains. Mol. Biosyst. 8, 1528–1534 (2012)
8. Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., Kanehisa, M.: Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. Bioinformatic 24, i232–i240 (2008)
9. Yamanishi, Y., Kotera, M., Kanehisa, M., Goto, S.: Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. Bioinformatic 26, i246–i254 (2010)
10. Wishart, D.S., Knox, C., Guo, A.C., Shrivastava, S., Hassanali, M.: DrugBank: a comprehensive resource for in silico drug discovery and exploration. Nucleic. Acids. Res. 34, D668–D672 (2006)
11. Kuhn, M., Szklarczyk, D., Franceschini, A., et al.: STITCH 3: zooming in on protein-chemical interactions. Nucleic. Acids. Res. 40, D876–D880 (2012)
12. Gregori-Puigjane, E., Setola, V., Hert, J., Crews, B.A., Irwin, J.J., et al.: Identifying mechanism-of-action targets for drugs and probes. Proc. Natl. Acad. Sci. U S A 109, 11178–11183 (2012)
13. Rask-Andersen, M., Almen, M.S., Schioth, H.: Trends in the exploitation of novel drug targets. Nat. Rev. Drug. Discov. 10, 579–590 (2011)
14. Hopkins, A.L., Groom, C.R.: The druggable genome. Nat. Rev. Drug. Discov. 1, 727–730 (2002)
15. Zhao, X.M., Iskar, M., Zeller, G., Kuhn, M., van Noort, V., Bork, P.: Prediction of drug combinations by integrating molecular and pharmacological data. PLoS. Comput. Biol. 7, e1002323 (2011)
16. Wang, Y.L., Xiao, J.W., Suzek, T.O., et al.: PubChem: a public information system for analyzing bioactivities of small molecules. Nucleic. Acids. Res. 37, W623–W633 (2008)
17. Apaweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., et al.: UniProt: the Universal Protein knowledgebase. Nucleic. Acids. Res. 32, D115–D119 (2004)
18. Michael, A., Catherine, A.B., Judith, A.B., David, B., Heather, B., et al.: Gene Ontology: tool for the unification of biology. Nat. Genet. 25, 25–29 (2000)
19. Kanehisa, M., Goto, S.: KEGG: kyoto encyclopedia of genes and genomes. Nucleic. Acids. Res. 28, 27–30 (2000)
20. Ge, X., Yamamoto, S., Tsutsumi, S., Midorikawa, Y., Ihara, S., Wang, S.M., Aburatani, H.: Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. Genomics 86, 127–141 (2005)
21. Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., Willighagen, E.: The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. J. Chem. Inf. Comput. Sci. 43, 493–500 (2003)
22. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197 (1981)
23. Zhao, X.M., Li, X., Chen, L., Aihara, K.: Protein classification with imbalanced data. Proteins 70, 1125–1132 (2008)
24. Zhu, F., Shi, Z., Qin, C., Tao, L., Liu, X., Xu, F., et al.: Therapeutic target database update 2012: a resource for facilitating target-oriented drug discovery. Nucleic. Acids. Res. 40, D1128–D1136 (2012)

Using Predictive Models to Engineer Biology: A Case Study in Codon Optimization

Alexey A. Gritsenko[1,2,3], Marcel J.T. Reinders[1,2,3], and Dick de Ridder[1,2,3]

[1] The Delft Bioinformatics Lab, Department of Intelligent Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[2] Platform Green Synthetic Biology, P.O. Box 5057, 2600 GA Delft, The Netherlands
[3] Kluyver Centre for Genomics of Industrial Fermentation, P.O. Box 5057, 2600 GA Delft, The Netherlands

Abstract. Given recent advances in synthetic biology and DNA synthesis, there is an increasing need for carefully engineered biological parts (e.g. genes, promoter sequences or enzymes) and circuits. However, forward engineering approaches are thus far rarely used in biology due to lack of detailed knowledge of the biological mechanisms. We describe a framework that enables forward engineering in biology by constructing models predictive of properties of interest, then inverting and using these models to design biological parts.

We demonstrate the applicability of the proposed framework on the problem of codon optimization, concerned with optimizing gene coding sequences for efficient translation. Results suggest that our data-driven codon optimization (DECODON) method simultaneously considers the effects multiple translation mechanisms to produce optimal sequences, in contrast to existing codon optimization techniques.

Keywords: synthetic biology, codon optimization, support vector regression, genetic algorithms.

1 Introduction

In biotechnology, microorganisms such as yeast are genetically engineered for improved production of foods, beverages, fuels and pharmaceuticals. Recent advances in synthetic biology and dropping cost of DNA synthesis have led to a growing need for methods to engineer biological parts (promoter regions, gene *coding sequences* (CDSs) and even entire enzymes) with specific properties. Whereas in many engineering disciplines optimization techniques are routinely used to design such parts (e.g. aircraft wings [16]), in synthetic biology this is not yet the case. This stems from a lack of fundamental biological knowledge on the processes in which these parts are involved.

For some problems, this limitation can be overcome by constructing predictive models for properties of biological parts (e.g. promoter strength, mRNA translation rate or enzyme activity) and inverting the constructed models to design biological parts with desired properties. A successful use of such a "black-box"

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 159–171, 2013.

modeling approach would enable forward engineering in areas of biology where detailed knowledge of the underlying processes is unavailable. We showcase the use of our proposed framework on the problem of codon optimization, in which a gene coding sequence is changed to obtain a desired translation rate of the mRNA into protein while keeping the amino acid sequence intact.

The degeneracy of the genetic code manifests itself in the differential use of synonymous codons in different organisms and different genes in the same organism. It has been long noticed that organisms preferentially use just one or two codons out of a family of codons translated into the same amino acid. This preference, termed *codon usage bias* (CUB), is more pronounced in highly expressed genes, which sometimes exclusively use only the preferred codons. For this reason it is believed that in unicellular organisms, such as baker's yeast *Saccharomyces cerevisiae* and the bacterium *Escherichia coli*, the codon bias of a gene is related to its translation rate [1]. Over the years numerous methods (called *indices*) summarizing the degree of CUB of a gene in a single number have been proposed and have been demonstrated to correlate with intracellular mRNA and protein levels [3].

These correlations have been used in a process called *codon optimization* to modify gene CDSs such that their translation rate is maximized, by introducing synonymous codon substitutions which increase one of the codon indices [9]. Codon optimization is routinely applied in biotechnology to overexpress genes for heterologous protein production and heterologous pathway expression [13]. However, CUB only partially explains the difference in translation rates among genes. Although the precise mechanisms influencing gene translation rates are not known, there is evidence suggesting that codon pair usage, tRNA recycling [2], mRNA secondary structure [19], adaptation to an organisms tRNA pool, mRNA untranslated regions (UTRs) and protein amino acid charge [19] may influence translation initiation and elongation rates. The relative influence of these factors on translation is not understood, making it difficult to combine them in a single codon optimization strategy. To our knowledge only Maertens et al. [15] have successfully combined multiple codon optimization objectives, by equally weighting them.

We present DECODON (data-driven codon optimization), an approach to codon optimization that combines multiple optimization objectives in a data-driven way by constructing a regression model. We use *Support Vector Regression* (SVR) [7] to predict *ribosome density*, a measure related to translation rate, based on coding sequence features of *S.cerevisiae* genes. We then invert this predictor by using it inside a genetic algorithm to optimize gene CDSs for desired ribosome density.

2 Materials and Methods

2.1 Dataset

To our knowledge no datasets with direct measurements of translation rates are available. However, Ingolia et al. [11] performed genome-scale measurements of

average *ribosome density*, defined as the number sequencing reads originating from parts of mRNA molecules covered by ribosomes in all mRNA copies of a particular gene, divided by the length of the gene transcript. Ribosome density is indicative of translation rate, as genes with higher densities are expected to produce more protein per copy of mRNA.

The number of gene mRNA copies per cell depends on its transcription rate and the stability of its mRNA. Although the relationship is poorly understood, the latter may be influenced by the secondary structure of the mRNA, which can differ between synonymous (i.e. encoding the same peptide) versions of a gene. In order to take the potential influence of coding sequence on the transcript levels into account, we propose to directly (i.e. without normalizing by the mRNA *read density*) use ribosome density as a measure of gene translation rate.

Yeast gene CDSs were obtained from the Saccharomyces Genome Database and the matching 5′- and 3′-UTR sequences were obtained from Nagalakshmi et al. [17] and Yassour et al. [21] (preference given to the former in cases when the two studies were not in agreement). The resulting dataset contains 5,048 yeast genes, each associated with coding and UTR sequences and a measured ribosome density.

2.2 Sequence Features

In order to construct a predictor of ribosome density from gene sequences a number of candidate sequence-based features identified from the literature have been computed for each gene in the dataset. These features were then used in a multivariate regression training step. Selected candidate features (Table 1) include a subset of existing codon bias indices (13 features); protein indices and protein properties (12 features); and nucleotide, codon and amino acid composition features (122 features). Prior to training, features as well as the ribosome density to be predicted were standardized to zero mean and unit variance.

2.3 Regression Model Training

ϵ-SVR [4] has been chosen as a regression method as it supports nonlinear regression through the use of kernels, allowing for complex models, and because efficient training algorithms are available. SVR relies on the choice of several parameters, including the cost parameter C, the error in sensitivity ϵ, the regression kernel and its parameters. Often, due to the lack of a theoretical framework for choosing these parameters, a grid search approach is used to find a combination of parameters that minimizes the regression error. This training procedure, if performed inside cross-validation (CV), becomes computationally very expensive.

As a performance measure we calculate the coefficient of determination R^2. Normally this measure approaches 1 with increasing model complexity regardless of its validity and is therefore not suitable for assessing quality of complex (nonlinear, many features) models. However, if the coefficient of determination is computed using CV (denoted R^2_{CV}), it becomes a measure of the amount of variance in *unseen* data explained by the model. Similar to the coefficient of

Table 1. Sequence-based features used as initial input for regression model training. CF and SF respectively stand for the number of candidate features in the feature group and the number of features selected for the final ribosome density predictor. Description of codon indices can be found in Cannarozzi and Schneider [3].

Name	Description	SF	CF
CAI	*Codon Adaptation Index* measures the extent to which a gene is composed of codons from the highly expressed genes.	0	1
tAI	*tRNA Adaptation Index* measures the extent to which a gene consists of codons recognized by abundant tRNAs. It is computed for the full CDS and its first 14, 17 and 19 codons (tAI, tAI_{14}, tAI_{17} and tAI_{19} respectively) [19].	3	4
N_c	*Effective number of codons* estimates the number of uniformly used codons that would produce the CUB observed in a gene.	0	1
D_{ncu}	*Distance to native codon usage* [18] measures the difference between codon usage of a gene and the overall codon usage of the organism.	1	1
E_w	*Weighted sum of relative entropy* measures the degree of deviation from equal usage of synonymous codons using the Shannon entropy.	1	1
CPB	*Codon Pair Bias* score [5] is computed as the sum of log-ratios of observed and expected codon pair counts.	0	1
TPI_2	*tRNA Pairing Index* measures the extent of potential tRNA re-use during gene translation.	1	1
F_{op}	For computing the *Frequency of optimal codons*, optimal codons were chosen as corresponding to the most abundant tRNA species.	1	1
RCBS	*Relative codon usage bias* measures codon usage difference of a gene with respect to the its nucleotide composition.	0	1
P_1	Mean number of non-specific tRNA interactions per elongation cycle.	1	1
prot	Protein hydrophobicity, aromaticity, aliphatic and instability indices.	3	4
Q_{port}	Protein net charge, isoelectric point and weight.	3	3
Q_{side}	Mean amino acid side chain charge computed for the full protein and its first 4, 11, 15 and 40 amino acids [19].	0	5
len	Lengths of the CDS, the 5′- and the 3′-UTR regions.	3	3
nuc	Nucleotide and dinucleotide frequencies of the CDS regions.	7	20
GC_{15}	GC-content computed for the first 15 codons of the CDS	1	1
RSCU	*Relative Synonymous Codon Usage* is computed for each codon (except ATG) as the ratio between the observed number of its occurrences and the mean number of occurrences for codons encoding the same amino acid.	41	63
codon^2	tAI and CAI weights of the second codon in the CDS (denoted tAI^2 and CAI^2).	2	2
amino	Amino acid frequencies.	6	21
ΔG	Gibson free energy for mRNA secondary structures predicted by the Vienna RNA package [10]. It is computed for the 5′-/3′-UTR sequences; and the first 17, 34, and 53 codons of the CDS [19] with ($\Delta G_{\text{5′-UTR,CDS}_{17}}$, $\Delta G_{\text{5′-UTR,CDS}_{34}}$ and $\Delta G_{\text{5′-UTR,CDS}_{53}}$) and without ($\Delta G_{\text{CDS}_{17}}$, $\Delta G_{\text{CDS}_{34}}$ and $\Delta G_{\text{CDS}_{53}}$) 5′-UTR sequence	4	12

determination computed without CV, the cross-validation R^2_{CV} approaches 1 as *generalization* becomes better, but can be negative if the trained model explains less variance in unseen data than a constant model. We believe that R^2_{CV} is a suitable measure for assessing quality of nonlinear models and use it to optimize and assess performance of our regression models.

Parameter Preselection: To keep the amount of computation tractable, we first *screened* the parameter space by training predictors with different parameter settings and assessing their coefficient of determination computed by 10-fold CV ($R^2_{10\mathrm{CV}}$) on the complete dataset (Figure 1, block A). Screening results (data not shown) indicated that the performance of RBF and polynomial kernels on the considered dataset is comparable, which led us to consider only polynomial kernels $K(u, v) = (\gamma \cdot \langle u, v \rangle + 1)^d$ with degrees $d = 2, 3, 4$ for the actual parameter selection stage. Based on the screening $R^2_{10\mathrm{CV}}$ results, ranges for parameters C, γ and ϵ were set to $\{1\} \cup \{0.001 \cdot 3^i\}$ for $i = 0, \ldots, 6$.

Parameter Estimation: The preselected parameter ranges were used to estimate optimal SVR parameter settings (Figure 1, block B) in a grid search procedure. For each combination of parameters an SVR is trained and its $R^2_{4\mathrm{CV}}$ is computed to select a *single* combination of SVR parameter settings with the

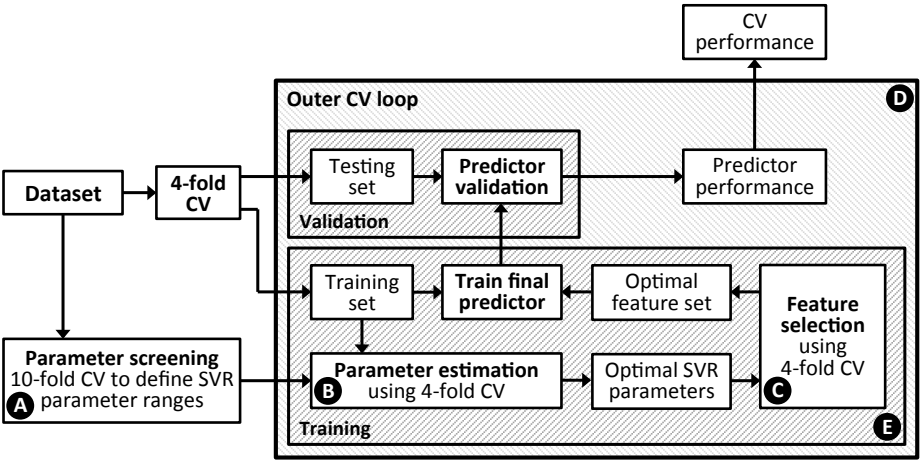


Fig. 1. Predictor training and evaluation scheme (adapted from [20]). The full dataset is used to preselect SVR parameter ranges (block A) and evaluate the training protocol using CV (block D). Predictor training consists of parameter estimation (block B) used to find an optimal set of SVR parameters, for which feature selection is performed (block C). The optimal parameters and the selected features are used to train the final predictor which is evaluated on the testing set of the CV loop. The same training procedure (block E) is used to train the final predictors used for sequence optimization on the *complete* dataset.

best performance. This combination is then used in the subsequent feature selection step.

Feature Selection: Feature selection was used to eliminate features that do not contribute to the model's generalization capability. This also allowed for selecting a concise set of features which can be interpreted biologically. While generally yielding good results, wrapper approaches to feature selection are computationally very demanding. To lower the computational load, backward feature elimination [12] was performed only on the SVR parameter settings obtained as discussed above (Figure 1, block C). At every step of the feature elimination procedure, given n features, we computed $R^2_{4\mathrm{CV}}$ for n predictors trained on subsets of $n-1$ features (i.e. obtained by removing one of the features). A subset with the highest R^2_{CV} was then selected for the next step of the feature elimination procedure. After the procedure was complete, the number of features (and the corresponding subset) with the best performance was chosen. If multiple subsets gave optimal performance, the smallest one was selected. The selected features were used to train the final predictor on the available data (Figure 1, block E).

Training Strategy Evaluation: In order to obtain an unbiased estimate of the predictor performance we used a second 4-fold CV loop (Figure 1, block D) around the described parameter estimation and feature selection strategies. The $R^2_{4\mathrm{CV}}$ values computed in the outer CV loop are reported in Section 3 as estimates of predictor generalization.

2.4 Sequence Optimization

In order for the constructed predictor $y = f(x)$ to be useful for sequence optimization, it first needs to be "inverted" such that it can be used to find sequences x that have the desired ribosome density $\check{y}$. Constructing the inverse function $x = f^{-1}(y)$ for SVR is impossible. Moreover, solving this function for a given $\check{y}$ would yield multiple nonsynonymous sequences x, thereby presenting an additional problem of selecting the suitable sequences from a large pool of solutions. Instead we implicitly invert the predictor by searching through the space of sequences x_i synonymous to the original sequence x to find $\check{x}$ such that its predicted ribosome density $f(\check{x})$ is close to the desired $\check{y}$.

Genetic Algorithm: The space of all nucleotide sequences synonymous to a given sequence x grows exponentially with the length of the sequence. Typically, it is too large to evaluate all possible x_i and requires an efficient search strategy to find (an approximation of) $\check{x}$ in a timely manner. *Genetic Algorithms* (GAs), specifically tailored for large discrete optimization problems, use computational equivalents of genetic crossover, mutation and selection concepts from biological systems to evolve a pool of potential solutions to a given optimization problem. The problem of finding an $\check{x}$ whose predicted ribosome density $f(\check{x})$ is as close

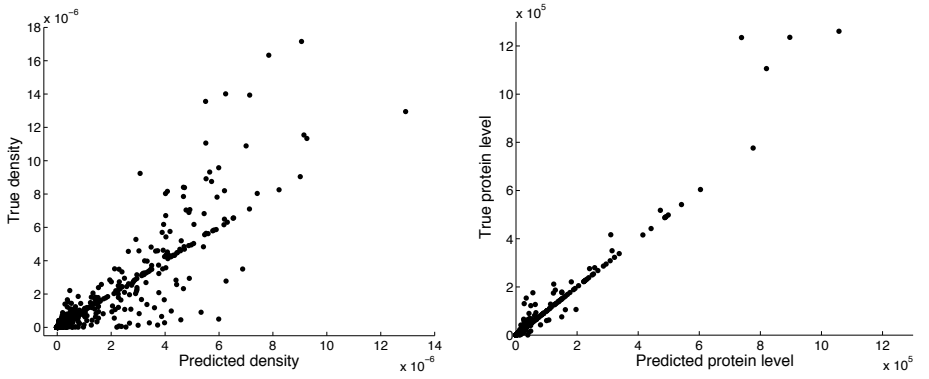
as possible to a desired level $\check{y}$ can be cast into an optimization problem and tackled using GAs if $g(x) = |f(x) - \check{y}|$ is used as an objective to be minimized. In practical applications, optimized gene sequences are synthesized and cloned into living cells in the wet lab. It is then required that the sequences do not contain certain motifs, such as restriction sites of enzymes used in cloning. This presents an optimization constraint that has to be taken care of by the GA. Treating this constraint as an additional objective of minimizing the number of undesired motifs present in the sequence allows to refrain from banning parts of the search space at the cost of casting the problem of finding $\check{x}$ into a multi-objective discrete optimization problem with two objectives. If it exists, the solution to the original problem will then be among the non-dominated solutions (i.e. solutions that cannot be improved in both objectives simultaneously) of the multi-objective optimization problem.

NSGA-II [6], a multi-objective GA, was chosen to solve the optimization problem as previously it has been successfully applied to DNA sequence optimization. It was implemented using multi-point crossover with a rate of 0.9; a mutation operator synonymously changing every sequence codon with probability $\frac{1}{n}$, where n is the number of degenerate codons in the sequence; and a binary tournament selection operator. For the genes optimized in this paper, the number of crossover points was set to 100.

3 Results

3.1 Regression Model

The cross-validation loop used to evaluate the regressor training strategy described in Section 2.3 gave an $R^2_{4\mathrm{CV}} = 0.66 \pm 0.03$, suggesting that the proposed



(a) Ribosome density predictor. (b) Protein level predictor.

Fig. 2. Predicted vs. true (a) ribosome density and (b) protein level plotted for *S.cerevisiae* genes.

strategy produces regressors that generalize well on unseen data. This strategy was employed to train the *final* ribosome density predictor (shown in Figure 2(a)) for use in codon optimization on the complete dataset.

Selected Features: The final predictor contained 78 features (Table 1, Figure 3), including codon indices, protein features, sequence composition and mRNA structure features selected to best explain the data. While black-box predictors are generally hard to interpret in biological terms, the fact that a certain feature was selected in the final predictor suggests that the mechanism it describes could indeed be used by the translation machinery. In this way, selection of the tRNA Pairing Index (TPI_2) suggests presence in yeast of a tRNA recycling mechanism, in which outgoing tRNA molecules stay bound to the ribosome to be recharged and reused in the course of translation [3]. Selection of the CAI^2 and tAI^2 features, describing respectively the extent to which the second codon of a gene is used in highly expressed genes of *S.cerevisiae* and its adaptation to the organisms tRNA pool, suggests that choice of the second codon influences ribosome density. Fredrick and Ibba [8] observe that the second codon is usually a highly frequently used codon that is translated more quickly, and speculate that this mechanism may be required for efficient recycling of the initiator tRNA.

Similarly, the selected tAI_{17}, tAI_{19}, and the $\Delta G_{5'\text{-UTR,CDS}_{17}}$, $\Delta G_{5'\text{-UTR,CDS}_{53}}$ and $\Delta G_{\mathrm{CDS}_{53}}$ features suggest that the mechanism of slowly translated "ramp" in the beginning of the CDS [19] influences gene translation rate. It is believed that the role of this "ramp" is to generate space between translating ribosomes and thereby prevent ribosome collision [8, 19]. The same mRNA structure features

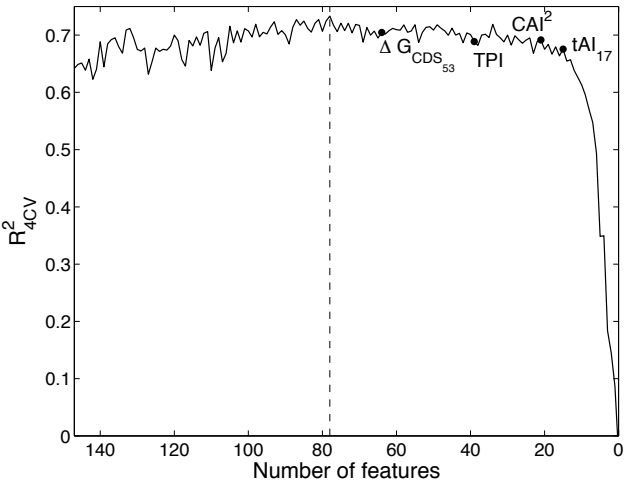


Fig. 3. Cross-validated $R^2_{4\mathrm{CV}}$ for the backward feature elimination procedure during final predictor training. Features eliminated at a particular step are marked with black circles. The maximum $R^2_{4\mathrm{CV}}$ is achieved at 78 features (see Table 1).

also describe the accessibility of the 5′-UTR for translation initiation by the ribosome machinery, suggesting it as another *S.cerevisiae* mechanism influencing gene translation.

3.2 Codon Optimization

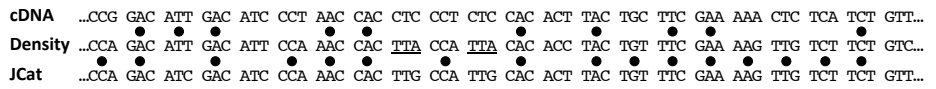
The final ribosome density predictor (Section 3.1) was used to optimize sequences of the genes 4CL (*4-coumaric acid-CoA ligase*, 562 codons) and PAL1 (*phenylalanine ammonia lyase*, 726 codons) involved in flavonoid biosynthesis [13]. The genes' cDNA, obtained from the plant *Arabidopsis thaliana*, was optimized using the described GA for *maximum* ribosome density. Based on preliminary experiments, optimization was performed for 200 generations with a population size equal to the gene length in codons. An initial population was generated by backtranslating genes from their amino acid sequences by choosing codons with probabilities proportional to their CAI weights. The 5′- and 3′-UTR sequences were set based on the respective sequences of the GPD promoter and CYC1 terminator sequences used in the pAG416GPD yeast expression vector. The *SpeI* and *XhoI* restriction site sequences used for cutting the expression vector were treated as undesired motifs.

Table 2 shows that the predicted ribosome density of the optimized sequences is significantly higher than that of the plant cDNA. As a sanity check, we compared sequences optimized using our method DECODON to sequences optimized by JCat [9], a well-known codon optimization tool that optimizes sequences for high CAI. The constructed predictor also predicts a significant increase in ribosome density for the JCat-optimized sequences (Table 2), showing that the trained predictor agrees with the currently used codon optimization methods. Note that the predicted ribosome density for the DECODON-optimized sequences is nearly two-fold higher than that of the JCat-optimized sequences.

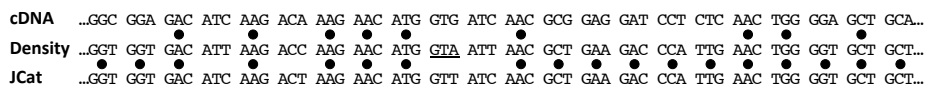
Sequence Analysis: Compared to the cDNA sequences, the DECODON- and JCat-optimized versions have roughly the same number of codon substitutions. To highlight the specific differences between the sequences, we compared them

Table 2. Sequence optimization results for the 4CL and PAL1 genes. Predicted ribosome densities are shown for the plant cDNA, sequences codon-optimized using JCat [9] and sequences optimized using DECODON. The number of different codons and the fold increase in the predicted density are computed relative to the cDNA sequences.

	4CL			PAL1		
Type	Different codons	Predicted density	Fold inc.	Different codons	Predicted density	Fold inc.
cDNA	N/A	0.0000000090	1	N/A	0.0000000524	1
JCat	338 (60.14%)	0.0000101491	1128	414 (57.02%)	0.0000079718	152
DECODON	361 (64.23%)	0.0000201560	2240	444 (61.16%)	0.0000172657	329



(a) 4CL gene, codons 40 to 61.



(b) PAL1 gene, codons 20 to 41.

Fig. 4. Comparison of part of the codon-optimized sequences (JCat and ribosome density optimized using DECODON). Matching codons are marked with black circles. Underscored codons are not explained by the "one amino acid - one codon" rule.

to each other. It can be seen from Figure 4 that codon usage in the DECODON sequences is more similar to that of the JCat-optimized genes than to that of the original sequences.

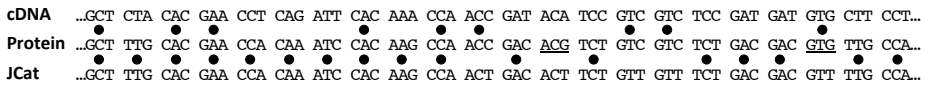
When optimized for maximum ribosome density, codon usage of the optimized sequences follows the "one amino acid - one codon" rule meaning that for each amino acid only a single (preferred) codon is used to encode it. The preferred codons in the genes optimized by DECODON mostly correspond to the codons with high CAI weights (the JCat- and density-optimized 4CL and PAL1 genes differ only in 126 and 150 codons respectively) with a few notable exceptions: (a) ACC is preferred for the amino acid threonine; (b) GTC is preferred for valine; (c) TGC is preferred for cysteine; and (d) ATT is preferred for isoleucine.

The preference rules account for all but a few codon differences (underscored in Figure 4) between the optimized sequences. These substitutions, when introduced in the sequences optimized using the "one amino acid - one codon rule", influence codon indices and mRNA features ($\Delta G_{\mathrm{CDS}_{53}}$ and $\Delta G_{5'\text{-UTR},\mathrm{CDS}_{53}}$), according to which the mRNA secondary structures at the 5′-UTR become less stable. This further suggests that the constructed predictor takes into account multiple translation mechanisms, even when used to optimize genes for maximum ribosome density.

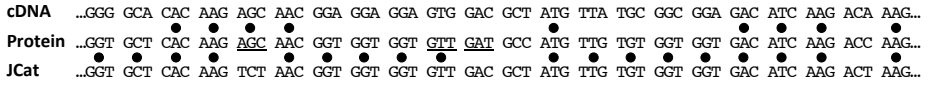
3.3 Applicability to Other Datasets

To demonstrate the applicability of the framework proposed in this paper to different datasets, we used it to optimize codon use based on the predicted absolute protein level measurements of 756 proteins [14]. All the training steps (parameter preselection, training strategy evaluation and final predictor training) were repeated, yielding an cross-validation $R^2_{4\mathrm{CV}} = 0.65 \pm 0.09$ and a final predictor with 138 features (Figure 2(b)). This large number of features, explained by the relatively high variance in the $R^2_{4\mathrm{CV}}$ used for feature selection due to the limited size of the dataset, hampers further biological interpretation.

The 4CL and PAL1 gene sequences optimized for maximum protein levels using the constructed predictor show a "one amino acid - one codon"' rule



(a) 4CL gene, codons 5 to 26.



(b) PAL1 gene, codons 5 to 26.

Fig. 5. Comparison of part of the codon-optimized sequences (JCat and absolute protein levels optimized using DECODON)

behavior similar to the density-optimized genes with several differences: (a) TGT is preferred for cysteine (as in JCat); (b) ATC is preferred for isoleucine (as in JCat); and (c) GCT and GCC are preferred for alanine. Similarly, these rules explain all but a few codon substitutions near to the $5'$ end of the CDS (Figure 5). The codon usage similarities between the protein- and density-optimized gene sequences show that the proposed framework can be applied to various types of biological data to enable forward engineering approaches. However, wet-lab experiments are required in order to determine which of the constructed predictors is better suited for codon optimization.

4 Discussion

We have described a generic framework for forward engineering of biological systems and demonstrated its use by optimizing genes for maximum ribosome density and maximum protein levels using predictors constructed from the corresponding yeast datasets. The general agreement between the optimized gene sequences obtained by us and gene sequences optimized using an existing codon optimization method suggests that the proposed approach can be successfully utilized for forward engineering of biological parts, whereas the differences between the sequences suggest that our codon optimization method DECODON simultaneously considers the effects of multiple translation mechanisms to produce optimal sequences. Time complexity of DECODON is much higher than that of JCat, however, it is negligible compared to the time involved in ordering and experimenting with the synthesized DNA.

Features selected for the final ribosome density predictor and the exceptions to the "one amino acid - one codon" rule in the optimized sequences show that data-driven models can combine multiple features describing (competing) biological mechanisms in a way that best explains the available data. While the effect of combining multiple mechanisms in a single predictor is hard to observe in sequences optimized for maximum ribosome density (or protein level), we believe that it would be more pronounced in sequences optimized for intermediate ribosome density, in which no one single mechanism would have a dominating influence.

Using black-box models for combining multiple (potential) mechanisms in a single predictor is particularly useful in areas where precise workings of a system are not known, but hypotheses on its important aspects can be generated and described by features. Note that a danger associated with the interpretation of the results is that the constructed model will select features that correlate with the property it is trained to predict, rather than the features describing the actual underlying mechanisms. For example, Qian et al. [18] suggest that strong CUB in highly expressed genes is not related to translation rate of those genes, but is rather a consequence of random mutations and the evolutionary pressure to keep codon usage and tRNA availability of an organism balanced. Nevertheless our models exhibit the "one amino acid - one codon" behavior when genes are optimized for maximum density/protein levels. It is, therefore, crucial to validate predictive models by testing their predictions in the wet-lab prior to their application.

For the constructed predictors (especially in the case of the protein level predictor) we observed that a single codon substitution often leads to changes in many features. These changes are often difficult to interpret and to link to the effect a substitution has on the prediction. Nevertheless, we believe that by trading interpretability for general applicability, our framework will enable forward engineering of various parts essential for synthetic biology such as promoters, coding sequences and UTRs.

References

[1] Angov, E.: Codon usage: Nature's roadmap to expression and folding of proteins. Biotechnology Journal 6(6), 650–659 (2011)
[2] Cannarozzi, G., Schraudolph, N.N., Faty, M., von Rohr, P., Friberg, M.T., Roth, A.C., Gonnet, P., Gonnet, G., Barral, Y.: A role for codon order in translation dynamics. Cell 141, 355–367 (2010)
[3] Cannarozzi, G.M., Schneider, A.: Codon evolution: mechanisms and models. OUP Oxford (2012)
[4] Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 1–27 (2011)
[5] Coleman, J.R., Papamichail, D., Skiena, S., Futcher, B., Wimmer, E., Mueller, S.: Virus attenuation by genome-scale changes in codon pair bias. Science 320(5884), 1784–1787 (2008)
[6] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[7] Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems, pp. 155–161 (1997)
[8] Fredrick, K., Ibba, M.: How the sequence of a gene can tune its translation. Cell 141(2), 227–229 (2010)
[9] Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D.C., Jahn, D.: JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. Nucleic Acids Research 33(suppl. 2), 526–531 (2005)

[10] Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast folding and comparison of rna secondary structures. Monatshefte für Chemie/Chemical Monthly 125(2), 167–188 (1994)
[11] Ingolia, N.T., Ghaemmaghami, S.A., Newman, J.R.S., Weissman, J.S.: Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. Science 324(5924), 218–223 (2009)
[12] Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1), 273–324 (1997)
[13] Koopman, F., Beekwilder, J., Crimi, B., van Houwelingen, A., Hall, R.D., Bosch, D., van Maris, A.J.A., Pronk, J.T., Daran, J.-M.: De novo production of the flavonoid naringenin in engineered *Saccharomyces cerevisiae*. Microbial Cell Factories 11(1), 155 (2012)
[14] Lu, P., Vogel, C., Wang, R., Yao, X., Marcotte, E.M.: Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. Nature Biotechnology 25(1), 117–124 (2006)
[15] Maertens, B., Spriestersbach, A., von Groll, U., Roth, U., Kubicek, J., Gerrits, M., Graf, M., Liss, M., Daubert, D., Wagner, R., et al.: Gene optimization mechanisms: A multi-gene study reveals a high success rate of full-length human proteins expressed in *Escherichia coli*. Protein Science 19(7), 1312–1326 (2010)
[16] Mohammadi, B., Pironneau, O.: Shape optimization in fluid mechanics. Annu. Rev. Fluid Mech. 36, 255–279 (2004)
[17] Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., Snyder, M.: The transcriptional landscape of the yeast genome defined by rna sequencing. Science 320(5881), 1344–1349 (2008)
[18] Qian, W., Yang, J.R., Pearson, N.M., Maclean, C., Zhang, J.: Balanced codon usage optimizes eukaryotic translational efficiency. PLoS Genetics, 8(3), e1002603 (2012)
[19] Tuller, T., Veksler-Lublinsky, I., Gazit, N., Kupiec, M., Ruppin, E., Ziv-Ukelson, M.: Composite effects of gene determinants on the translation speed and density of ribosomes. Genome Biology 12(11), R110 (2011)
[20] Wessels, L.F.A., Reinders, M.J.T., Hart, A.A.M., Veenman, C.J., Dai, H., He, Y.D., Van't Veer, L.J.: A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics 21(19), 3755–3762 (2005)
[21] Yassour, M., Kaplan, T., Fraser, H.B., Levin, J.Z., Pfiffner, J., Adiconis, X., Schroth, G., Luo, S., Khrebtukova, I., Gnirke, A., Nusbaum, C., Thompson, D.-A., Friedman, N., Regev, A.: *Ab initio* construction of a eukaryotic transcriptome by massively parallel mRNA sequencing. Proceedings of the National Academy of Sciences 106(9), 3264–3269 (2009)

Active Learning for Protein Function Prediction in Protein-Protein Interaction Networks

Wei Xiong[1], Luyu Xie[1], Jihong Guan[2], and Shuigeng Zhou[1]

[1] School of Computer Science, and Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai, China
{wxiong,10300240052,sgzhou}@fudan.edu.cn
[2] Department of Computer Science & Technology, Tongji University, Shanghai, China
jhguan@tongji.edu.cn

Abstract. The high-throughput technologies have led to vast amounts of protein-protein interaction (PPI) data, and a number of approaches based on PPI networks have been proposed for protein function prediction. However, these approaches do not work well if annotated proteins are scarce in the networks. To address this issue, we propose an active learning based approach that uses graph-based centrality metrics to select proper candidates for labeling. We first cluster a PPI network by using the spectral clustering algorithm and select some proper candidates for labeling within each cluster, and then apply a collective classification algorithm to predict protein function based on these annotated proteins. Experiments over two real datasets demonstrate that the active learning based approach achieves better prediction performance by choosing more informative proteins for labeling. Experimental results also validate that betweenness centrality is more effective than degree centrality and closeness centrality in most cases.

Keywords: Protein function prediction, Active learning, Collective classification, Protein-protein interaction network.

1 Introduction

In recent years, the rapid development of high-throughput experimental biology has led to huge amounts of unannotated protein sequences. Meanwhile, experimentally determining protein function is expensive and time-consuming. So there is a wider and wider gap between the pace of discovery of protein sequences and that of functional annotation of known proteins. Therefore, protein function prediction has been a fundamental challenge of biology in the post-genomic era. Although many efforts have been made to solve this problem, the proportion of annotated proteins is still very low. Among the 13 million protein sequences, there are only 1% sequences having experimentally-validated annotations [1]. Even for the most well-studied model organisms, taking yeast as an example, approximately one-fourth of the proteins have no annotated functions [2].

Due to high cost and long duration of experimentally annotating protein function, there is increasing research on using computational approaches to predict

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 172–183, 2013.

protein function. The recent advent of high-throughput experimental biology has generated vast amounts of protein-protein interaction (PPI) data, which are represented as networks, where a node corresponds to a protein and an edge corresponds to an interaction between a pair of proteins. Thus, a number of prediction approaches based on PPI networks have been proposed. These approaches make use of the observation that proteins with short distance to each other in a PPI network are more likely to have similar functions.

However, current network-based approaches will fail to work when annotated proteins are scarce. To address this issue, in this paper we propose an active learning [3] based approach that uses graph-based centrality metrics to select good candidates for labeling. Our approach consists of two steps: we first cluster a PPI network by using spectral clustering algorithm and select proper candidates for labeling within each cluster, and then apply a collective classification algorithm to predict protein function based on these annotated proteins. To the best of our knowledge, this is the first study where active learning is employed to predict protein functions in PPI networks. The key idea behind active learning is that a machine learning algorithm can achieve higher accuracy with fewer training labels if it is allowed to choose the proper data for labeling from which it learns. Therefore, we let the learning algorithm pick a set of unannotated proteins to be labeled by an oracle (*i.e.*, a lab experiment), which will then be used as the labeled data set. In other words, we let the learning algorithm tell us which proteins to label, rather than select them randomly.

We conduct experiments on the *S.cerevisiae* and *M.musculus* functional annotation datasets, The experimental results show that the active learning based approach achieves better prediction performance by choosing more informative proteins for labeling. Experimental results also validate that betweenness centrality is more effective than degree centrality and closeness centrality in most cases. The rest of this paper is organized as follows: Section 2 presents our approach, Section 3 gives the experimental evaluation results, Section 4 describes related work, and finally Section 5 concludes the paper.

2 Method

2.1 Notation and Problem Definition

In this paper, a PPI network is represented as an indirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = (V_1, ..., V_n)$ is a set of n vertices and $\mathcal{E}$ is a set of weighted edges. Each vertex $V_i \in \mathcal{V}$ represents a protein and each edge $E_{i,j} \in \mathcal{E}$ represents an interaction between proteins V_i and V_j. Edge $E_{i,j}$ is labeled with a weight $w_{i,j}$ that indicates the interaction confidence. $\mathcal{F} = (F_1, ..., F_m)$ is the set of m functions assigned to the proteins, and each vertex $V_i \in \mathcal{V}$ is assigned with at least one function. The functions of vertex $V_i \in \mathcal{V}$ are denoted by

$$\Phi(V_i) = [f_{i,1}, f_{i,2}, ..., f_{i,j}, ..., f_{i,m}]^T \quad (1)$$

where

$$\begin{cases} f_{i,j} = 1, & if\ V_i\ has\ the\ function\ F_j; \\ f_{i,j} = 0, & otherwise. \end{cases} \tag{2}$$

$\mathcal{V}$ can further be divided into two sets: $\mathcal{X}$ — the labeled vertices and $\mathcal{Y}$ — the vertices whose functions need to be determined.

In this paper, our goal is to label as few vertices $\{Y_i\} \subset \mathcal{Y}$ as possible with at least one of the functions in $\mathcal{F}$ based on the available information of the corresponding PPI network, so that the labeled vertices $\{Y_i\}$ and $\mathcal{X}$ together constitute the training set, which can be used to train an as good as possible classifier. Here, active learning is used for data selection to be labeled, the collective classification method is employed for classifier training.

2.2 Active Learning Strategies for Protein Function Prediction

As we point out above, experimentally annotating protein function is expensive in terms of cost and effort, and current network-based approaches do not work well if annotated proteins are scarce. Therefore, strategies that minimize the amount of labeled data required in the supervised learning task would be useful. Active learning attempts to overcome the labeling bottleneck by asking queries in the form of unlabeled instances to be labeled by an oracle (*i.e.*, a lab experiment). In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. The key idea behind active learning is that a machine learning algorithm can achieve higher accuracy with fewer training labels if it is allowed to choose the most proper data for labeling from which it learns.

In this study, the PPI network is represented as a graph, so it seems reasonable that we leverage graph structure to identify the nodes (proteins) in the graph that are important (central) for labeling. That is, we expect that such central nodes are proper candidates to label. Furthermore, we also note that nodes of the same class tend to cluster together in the PPI network. This suggests that clustering the graph and then finding central nodes in the clusters my be a good way to find proper candidates. Therefore, we explore the spectral clustering algorithm to cluster the PPI network and then leverage graph-based centrality metrics to select central nodes in the clusters to label.

Under the active learning framework, there is a small set of labeled data and a large pool of unlabeled data available. A fixed number M of labels (usually called the *labeling budget*) is requested. Suppose that the selected nodes are distributed across the clusters of the PPI network, in proportion to the size of the cluster. Let n_i be the number of nodes in cluster C_i and N be total number of nodes in the PPI network. Then, m_i, the number of nodes to be selected from cluster C_i is given by

$$m_i = M * n_i/N \quad and \quad M = \sum_{i=1}^{K} m_i. \tag{3}$$

In each cluster C_i, m_i central nodes are selected to label. In what follows, we describe and discuss the spectral clustering algorithm and graph-based centrality metrics in detail.

Spectral Clustering Algorithm. Spectral clustering [4] is one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm. Detail description of the spectral clustering algorithm is presented as follows.

Given a PPI network, let $W \in \mathbb{R}^{n\times n}$ be its weighted adjacency matrix, $W_{ii} = 0$ and $W_{ij} = 0$ if the vertices V_i and V_j are not connected by an edge. The degree of a vertex $V_i \in \mathcal{V}$ is defined as

$$d_i = \sum_{j=1}^{n} W_{ij}. \tag{4}$$

Note that this sum only performs over all vertices adjacent to V_i, as for all other vertices V_j the weight W_{ij} is 0. The degree matrix D is defined as the diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal. The unnormalized graph Laplacian matrix is defined as

$$L = D - W. \tag{5}$$

Next, we compute the first k eigenvectors $u_1, ..., u_k$ of L, and let $U \in \mathbb{R}^{n\times k}$ be the matrix containing the vectors $u_1, ..., u_k$ as columns. For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U. Finally, we cluster the points y_i in $\mathbb{R}^k$ with the K-means algorithm into clusters $C_1, ..., C_k$.

Graph-Based Centrality Metrics. In this study, we consider three kinds of graph-based centrality metrics for active learning, including degree centrality, closeness centrality and betweenness centrality.

Degree centrality. Graph degree centrality is perhaps the simplest measure of centrality, it is defined as the number of links incident upon a vertex (*i.e.*, the degree of a vertex). So graph degree centrality of a vertex v is defined as follows:

$$C_D(v) = \deg(v). \tag{6}$$

Closeness centrality [5]. In connected graph there is a natural distance metric between all pairs of vertices, defined by the length of their shortest paths. The *farness* of a vertex is defined as the sum of its distances to all other vertices, and its *closeness* is defined as the inverse of the farness. Thus, the more central a vertex is the smaller its total distance to all other vertices. *Graph closeness centrality* measures how close a vertex is to all other vertices in the graph, it is defined as the inverse of the total distance to all other vertices:

$$C_C(v) = \frac{1}{\sum\limits_{t\in V} d(v,t)}. \tag{7}$$

where $d(v,t)$ is the distance from vertex v to vertex t in the graph. In unweighted graph, the distance is defined in terms of the number of edges that connect two vertices. And in weighted graph, we define the distance as the sum of weights of the edges that connect two vertices.

Betweenness centrality [6]. Graph betweenness centrality is perhaps one of the most prominent measures of centrality, it quantifies the number of times a vertex acts as a bridge along the shortest path between two other vertices. That is, vertices that have a high probability to occur on a randomly chosen shortest path between two randomly chosen vertices have a high betweenness. Graph betweenness centrality of a vertex v is evaluated as follows:

$$B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{8}$$

Above, σ_{st} is the total number of shortest paths from vertex s to vertex t and $\sigma_{st}(v)$ is the number of those paths that pass through v. As with closeness, we compute all shortest paths to get the centrality measure for all vertices.

2.3 Collective Classification: The Gibbs Sampling Approach

In this study, Gibbs sampling (GS) [7] is applied to predicting protein function. GS is one of the most commonly used collective classification algorithm that aims at finding the best label estimate for each un-annotated vertex $Y_i \in \mathcal{Y}$ by sampling each vertex label iteratively. Our approach consists of two steps: *bootstrapping* and *iterative classification*, the pseudo-code is illustrated in Algorithm 1. The details of the algorithm are presented in the following subsections.

Bootstrapping. According the observation that proteins with shorter distance to each other in the network are more likely to have similar functions, we use weighted voting to predict an initial functional probability distribution for a query protein (*i.e.* an un-annotated protein).

Given a query protein V_x, which has N_x neighbors, these corresponding edge weights can be represented as the vector as follows:

$$\mathcal{N}_x^w = [w_{x1}, w_{x2}, ..., w_{xi}, ..., w_{xN_x}]. \tag{9}$$

Then the probability of V_x having the j-th function F_j is computed as follows:

$$P_x^j = \frac{1}{Z_x^w} \sum_{i=1}^{N_x} w_{x,i} f_{i,j} \tag{10}$$

where Z_x^w is the normalizer:

$$Z_x^w = \sum_{j=1}^{m} \sum_{i=1}^{N_x} w_{x,i} f_{i,j}. \tag{11}$$

The larger the value of P_x^j is, the more likely protein V_x has the j-th function F_j. The initial functional probability distribution for query protein V_x is represented as an m-dimensional vector:

$$\boldsymbol{a_x} = [P_x^1, P_x^2, ..., P_x^m]. \tag{12}$$

Note that when predicting the functions of the query protein V_x, we consider only its labeled neighbor proteins. That is why we use $\mathcal{X} \cap \mathcal{N}_x^w$ in Algorithm 1 (Line 3), because unlabeled neighbor proteins can not be exploited in the bootstrapping step. This process is implemented in Alg. 1 from Line 2 to 4.

Iterative Classification. Iterative classification is performed in two steps:

- First, there is a fixed number B of iterations known as "burn-in" period. In this period, we only update $\boldsymbol{a_x}$ using weighted voting in each iteration. Corresponding codes of this period in Algorithm 1 are from Line 6 to 10.
- Second, there is a sampling period. In this period, not only do we update $\boldsymbol{a_x}$ in each iteration but we also maintain the count statistics as to how many times we have sampled the j-th function F_j for protein V_x. Codes corresponding to this period in Algorithm 1 are from Line 12 to 20.

Note that each protein can belong to one or more functions, therefore, we formulate protein functional annotation as a multiclass classification problem. More formally, the most likely function of protein V_x is computed like this:

$$b_x^1 = argmax_{j \in [1,m]} P_x^j \tag{13}$$

where b_x^1 is the value of j that maximizes the value of P_x^j, called the 1st-rank result. The second most likely function is denoted by b_x^2, called the 2nd-rank result. The third most likely function is denoted by b_x^3, called the 3rd-rank result, and so forth. In case that more than one element P_x^j has the same value, their ranks will be assigned randomly. For each protein V_x in the i-th iteration, an m-dimensional vector $\boldsymbol{b_{xi}}$ is created to record the ranking result:

$$\boldsymbol{b_{xi}} = [b_{xi}^1, b_{xi}^2, ..., b_{xi}^m]. \tag{14}$$

When the pre-specified number (threshold) S of iterations have elapsed, we get a matrix M_x with S rows and m columns for query protein V_x:

$$M_x = [\boldsymbol{b_{x1}}, \boldsymbol{b_{x2}}, ..., \boldsymbol{b_{xS}}]^T. \tag{15}$$

In the first column of the matrix M_x, the most frequently sampled function c_x^1 is regard as the first rank predicted function for the query protein V_x. In the second column of the matrix M_x, the most frequently sampled function c_x^2 excluding c_x^1 is regard as the second rank predicted function. In the third column of the matrix M_x, the most frequently sampled function c_x^3 excluding c_x^1 and c_x^2 is regard as the third rank predicted function, and so forth. Finally, we get an m-dimensional vector $\boldsymbol{c_x}$ for query protein V_x:

$$\boldsymbol{c_x} = [c_x^1, c_x^2, ..., c_x^m]. \tag{16}$$

Algorithm 1. Gibbs sampling based collective classification for protein function prediction in PPI networks.

```
// bootstrapping
for each query protein V_x do
    compute the initial a_x using X ∩ N_x^w
end for
// burn-in period
for i=1 to B do
    for each query protein V_x do
        update a_x using current assignments to N_x^w
    end for
end for
// sampling period
for i=1 to S do
    for each query protein V_x do
        update a_x using current assignments to N_x^w
        create b_xi to record the m-rank result
    end for
end for
for each query protein V_x do
    calculate the final result c_x based on matrix M_x
end for
```

3 Experimental Evaluation

3.1 Interaction and Annotation Data

We evaluate the performance of our approach with two functional annotation datasets. These two datasets are both based on Functional Catalogue (FunCat) annotation scheme [8] taken from Munich Information Center for Protein Sequences (MIPS)[1]. FunCat is organized as a hierarchically structured annotation system and consists of 28 main functional categories. FunCat annotations for S.cerevisiae are downloaded from Comprehensive Yeast Genome Database (CYGD) [9]. CYGD is a frequently used public resource for yeast related information. There are a total of 6168 proteins in the dataset, of which 4774 are annotated. These proteins belong to 17 functional categories. The second functional annotation dataset is Mouse functional Genome Database (MfunGD) [10]. MfunGD provides a resource for annotated mouse proteins and comprises 17643 annotated proteins. These annotated proteins belong to 24 functional categories.

In this study, protein interaction data is download from the STRING database [11], which is an integrated protein interaction database containing known and predicted protein interactions. These interactions were mainly derived from four data sources: genomic context, high-throughput experiments, conserved coexpression and previous knowledge. The most recent version of STRING covers about 5.2 million proteins from 1133 organisms.

We construct two protein interaction networks (one for S.cerevisiae and another for M.musculus) where a node corresponds to a protein and a weighted edge corresponds to an interaction between two proteins. Each node is assigned with at least one functional category and each edge is labeled with a weight

[1] `http://www.helmholtz-muenchen.de/en/ibis`

based on the interaction confidence. Proteins without interaction and annotation data are deleted. As a result, in the S.cerevisiae interaction network, there are 4687 proteins and 388846 interactions, and in the M.musculus interaction network there are 14277 proteins and 832128 interactions.

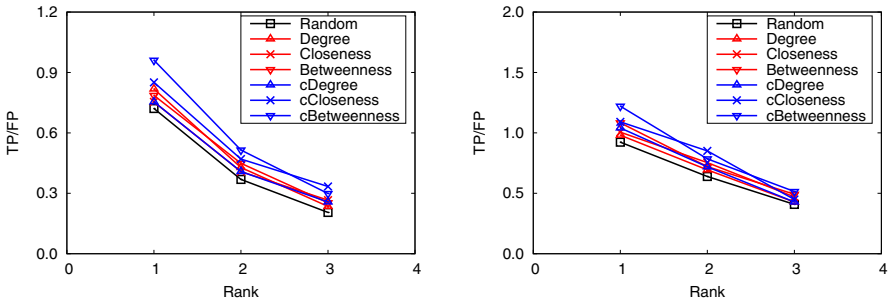
3.2 Experimental Methodology

In the experiments, we compare the performance of three kinds of data selection strategies. The first is *random data selection strategy* (baseline), which randomly selects nodes in the PPI network to label. The second is *graph structure based data selection strategy*, which leverages graph-based centrality metrics to select central nodes in the PPI network to label. The last is our proposed strategy, which first uses the spectral clustering algorithm to cluster the PPI network and then leverages graph-based centrality metrics to select central nodes in each cluster to label. Note that there are three kinds of graph-based centrality metrics (*degree centrality*, *closeness centrality* and *betweenness centrality*). Thus, in fact, we compare the performance of seven kinds of data selection strategies.

We set the proportion of annotated proteins to 5%, and for each data selection strategy, we run 20 experiments and report the average performance. In spectral clustering, we set the number of clusters K to 30 and 50 for S.cerevisiae and M.musculus respectively, this value is chosen by trial and error. As for collective classification, we set the burn-in period to 10 iterations (*i.e.* B=10) and collect 50 samples (*i.e.* S=50) in the sampling period. Since protein functional annotation is a multiclass classification problem, all competing methods calculate an m-rank predicted function vector $\boldsymbol{c_x}$ for each query protein V_x. In this setup, we define the i-th rank overall *true positive* (TP) as the number of proteins whose i-th rank predicted function c_x^i is one of the true functions of the protein V_x and the i-th rank overall *false positive* (FP) as the number of proteins whose i-th rank predicted function c_x^i is not one of the true functions of the protein V_x. Accordingly, as in [12] we use the ratio of TP/FP as the measure of performance, which depicts the relative magnitude between TP and FP.

3.3 Experimental Results

In the experiments, there are two PPI networks (corresponding to S.cerevisiae and M.musculus). For *S.cerevisiae*, the average number of functions that each protein has is 2.13, so we consider only the top 3 ($3=\lfloor 2.13 \rfloor + 1$) predictions. Fig. 1(a) shows the performance comparison of seven kinds of data selection strategies for the top-3 predictions. And for *M.musculus*, because the average number of functions that a protein possesses is 2.58, we consider also only the first 3 ($3=\lfloor 2.58 \rfloor + 1$) predictions. The results are shown in Fig. 1(b). In Fig. 1, for simplification, *Random* indicates the random data selection strategy; *Degree/Closeness/Betweeness* means the graph structure based strategy with the metric of *degree centrality/closeness centrality/betweenness centrality*; And *cDegree/cCloseness/cBetweeness* is our strategy with clustering plus the metric of *degree centrality/closeness centrality/betweenness centrality*.



(a) *S.cerevisiae* ($K = 30$) (b) *M.musculus* ($K = 50$)

Fig. 1. Performance comparison of seven kinds of data selection strategies

It can be seen from Fig. 1 that all the six graph structure based data selection strategies obtain more accurate predictions than the random data selection strategy, due to using graph-based centrality metrics to select central nodes in the PPI network to label. The results clearly show that the active learning based approach achieve a better prediction performance than the baseline approach. This means that given a similar number of labeled proteins, our active learning approach can achieve outstanding performance by choosing the most informative proteins to be labeled. We also notice that our proposed data selection strategies outperform other three graph structure based data selection strategies. As we explore the spectral clustering algorithm to cluster the PPI network before selecting protein candidates for labeling, this result shows that clustering is an important pre-processing step in active learning algorithm. The reason is that selecting candidates across clusters will make the distribution of selected candidates over different classes more balanced.

The experimental results also validate that using betweenness centrality as the graph-based centrality metric generally can achieve the best performance in most cases, which means betweenness centrality is more effective than degree centrality and closeness centrality. In addition, it is worth noting that higher rank functions are predicted better than lower ones, implying that the protein functions are well ranked by our approach.

4 Related Work

In a recent review [2], the existing network-based methods for protein function prediction are categorized into two main groups: direct methods and module-assisted methods. Direct methods propagate functional information through a PPI network and use the propagated information for functional annotation, examples include neighborhood counting methods and graph theoretic methods.

The majority method [13] and the indirect neighbors method [14] are two typical direct network-based approaches. Majority method [13] is the simplest direct method, it utilizes the biological hypothesis that interacting proteins probably

have similar functions, it ranks each candidate function based on the function's occurrences in the immediate neighbors. Indirect neighbors method [14] assumes that proteins interacting with the same proteins may also have some similar functions, It exploits both indirect and immediate neighbors to rank each candidate function. Functional flow method [15] is a graph theoretic method, it simulates a discrete-time flow of functions from all proteins. At every time step, the function weight transferred along an edge is proportional to the edge's weight and the direction of transfer is determined by the functional gradient.

Module-assisted methods first identify functional modules in the network and then assign functions to all the proteins in each module, representatives are hierarchical clustering-based method and graph clustering method. A key problem of this kind of methods is how to define the similarity between two proteins. Arnau *et al.* [16] used the shortest path between proteins as a distance measure and apply hierarchical clustering to detecting functional modules. Up to now, numerous graph-clustering algorithms have been applied to detecting functional modules, such as clique percolation [17] and edge-betweenness [18] clustering.

Additionally, Chua et al [19] presented a simple framework for integrating large amount of diverse information for protein function prediction by using simple weighting strategies and a local prediction method. Hu et al [20] hybridized the PPI information and the biochemical/physicochemical features of protein sequences to predict protein function. The prediction is carried out as follows: if the query protein has PPI information, the network-based method is applied; Otherwise, the hybrid-property based method is employed.

Active learning [3] is a form of supervised machine learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at some unlabeled data points. The key issue is to design the query strategy such that as few data points as possible are queried to achieve as large learning performance improvement as possible. The simplest and most commonly used query strategy is *uncertainty sampling* [21]. In this framework, an active learner queries the instance that the classifier is most uncertain. This strategy is often straightforward for probabilistic learning models. The *query-by-committee* (QBC) [22] strategy maintains a committee, each committee member is allowed to vote on the labelings of query candidates, the most informative query is considered to be the instance about which they most disagree. The fundamental premise behind the QBC strategy is minimizing the version space. The *expected model change* [23] strategy uses a decision-theoretic approach, it selects the instance that would impart the greatest change to the current model. The *expected error reduction* [24] strategy aims to measure not how much the model is likely to change, but how much its generalization error is likely to be reduced. It selects the instance that offer maximal expected error reduction to the classifier. The *density-weighted* [25] strategy suggests that the informative instances should not only be those which are uncertain, but also those which are representative of the underlying distribution.

Active learning has been applied to some bioinformatic problems, such as cancer classification [26], DNA microarray data analysis [27] and protein-protein

interaction prediction [28] etc. However, to the best of our knowledge, there is no work on active learning for protein function prediction in the literature.

5 Conclusion

In this study, we proposed an active learning based approach to conducting protein function prediction based on PPI networks. It first clusters a PPI network by using the spectral clustering algorithm and select some appropriate candidates for labeling within each cluster by using graph-based centrality metrics, and then applies a collective classification algorithm to predict protein function based on these annotated proteins. We conducted experiments on two real, publicly available PPI datasets. The experimental results show that the proposed active learning based approach, by choosing more informative proteins for labeling, achieves obviously better prediction performance than the baseline approach. Furthermore, betweenness centrality is more effective than degree centrality and closeness centrality in most cases.

Acknowledgments. This study was supported by China 863 Program (grant No. 2012AA020403), and NSFC (grants No. 61173118 and No. 61272380). Jihong Guan was also supported by the "Shuguang Scholar" Program of Shanghai Education Foundation.

References

1. Barrell, D., Dimmer, E., Huntley, R., Binns, D., O'Donovan, C., Apweiler, R.: The goa database in 2009 an integrated gene ontology annotation resource. Nucleic Acids Research 37, D396–D403 (2009)
2. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Molecular Systems Biology 3, 1–13 (2007)
3. Settles, B.: Active learning literature survey. University of Wisconsin, Madison (2010)
4. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
5. Sabidussi, G.: The centrality index of a graph. Psychometrika 31, 581–603 (1966)
6. Freeman, L.C.: A set of measures of centrality based upon betweenness. Sociometry 40, 35–41 (1977)
7. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in netwok data. AI Magazine 29, 93–106 (2008)
8. Ruepp, A., Zollner, A., Maier, D., Albermann, K., et al.: The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research 32, 5539–5545 (2004)
9. Güldener, U., Münsterkötter, M., Kastenmüller, G., Strack, N., et al.: Cygd: the comprehensive yeast genome database. Nucleic Acids Research 33, D364–D368 (2005)
10. Ruepp, A., Doudieu, O., Van den Oever, J., Brauner, B., et al.: The mouse functional genome database (mfungd): functional annotation of proteins in the light of their cellular context. Nucleic Acids Research 34, D568–D571 (2006)

11. Damian, S., Andrea, F., Michael, K., Milan, S., et al.: The string database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Research 39, D561–D568 (2011)
12. Bogdanov, P., Singh, A.K.: Molecular Function Prediction Using Neighborhood Features. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7, 208–217 (2010)
13. Schwikowski, B., Uetz, P., Fields, S.: A Network of Protein-Protein Interactions in Yeast. Nature Biotechnology 18, 1257–1261 (2000)
14. Chua, H.N., Sung, W.K., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. Bioinformatics 22, 1623–1630 (2006)
15. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. Bioinformatics 21(suppl. 1), i302-i310 (2005)
16. Arnau, V., Mars, S., Marin, I.: Iterative cluster analysis of protein interaction data. Bioinformatics 21, 364–378 (2005)
17. Adamcsek, B., Palla, G., Farkas, I.J., Derenyi, I., Vicsek, T.: Cfinder: locating cliques and overlapping modulesin biological networks. Bioinformatics 22, 1021–1023 (2006)
18. Dunn, R., Dudbridge, F., Sanderson, C.: The use of edge-betweenness clustering to investigate biological function in protein interaction networks. BMC Bioinformatics 6, 39 (2005)
19. Chua, H.N., Sung, W.K., Wong, L.: An efficient strategy for extensive integration of diverse biological data for protein function prediction. Bioinformatics 23(24), 3364–3373 (2007)
20. Hu, L., Huang, T., Shi, X., Lu, W., Cai, Y., et al.: Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties. PLoS ONE 6(1), e14556 (2011)
21. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1069–1078. ACL Press (2008)
22. Körner, C., Wrobel, S.: Multi-class ensemble-based active learning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 687–694. Springer, Heidelberg (2006)
23. Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. In: Advances in Neural Information Processing Systems, vol. 20, pp. 1289–1296. MIT Press (2008b)
24. Guo, Y., Greiner, R.: Optimistic active learning using mutual information. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 823–829. AAAI Press (2007)
25. Xu, Z., Akella, R., Zhang, Y.: Incorporating diversity and density in active learning for relevance feedback. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 246–257. Springer, Heidelberg (2007)
26. Liu, Y.: Active learning with support vector machine applied to gene expression data for cancer classification. Journal of Chemical Information and Computer Sciences 44(6), 1936–1941 (2004)
27. Vogiatzis, D., Tsapatsoulis, N.: Active learning for microarray data. International Journal of Approximate Reasoning 47(1), 85–96 (2008)
28. Mohamed, T.P., Carbonell, J.G., Ganapathiraju, M.K.: Active learning for human protein-protein interaction prediction. BMC Bioinformatics 11(suppl. 1), S57 (2010)

Conditional Random Fields for Protein Function Prediction

Thies Gehrmann[1], Marco Loog[2], Marcel J.T. Reinders[1,3,4], and Dick de Ridder[1,3,4]

[1] Delft Bioinformatics Lab, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[2] Pattern Recognition Lab, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[3] Netherlands Bioinformatics Centre, P.O. Box 9101, 6500 HB Nijmegen, The Netherlands
[4] Kluyver Centre for Genomics of Industrial Fermentation, P.O. Box 5057, 2600 GA Delft, The Netherlands

Abstract. Markov Random Fields (MRF) have been shown to be good predictors of functional annotation, using protein-protein interaction data. Many other sources of data can also be used in this prediction task, but they are typically not integrated. In this study, we extend a method using MRFs in order to allow the use of additional data.

A conditional random field (CRF) model is proposed as an alternative to an MRF model in order to remove the requirement of modeling relationships between the sources of data. We observe that a substantial performance improvement is possible using additional data, such as genetic interaction networks. The improvement gained from each source of evidence is not the same for each protein function, indicating that each source supplies different information. We demonstrate that CRFs can be used to efficiently integrate various sources of data to predict functional annotations.

1 Introduction

The annotation of genomes is of the utmost importance and the value a deeper understanding of biological processes has to the future of humanity can not be overstated. Proteins found on these genomes are assigned functional annotations based on biological experimentation. This is no easy task, since proteins may be involved in several kinds of functions, and testing every protein's involvement in every function is infeasible due to the number of proteins and the complexity of their interactions. Therefore, the functional annotation of proteins remains largely incomplete, even for the most well studied organisms. By predicting which proteins are most likely to have a specific function, we can reduce the expenses and work required to get a more complete genomic annotation. This is the reason why *in-silico* prediction of protein function is important.

Algorithms that predict functional annotations differ not only in their underlying method, but also in the data they operate upon. For a recent review,

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 184–195, 2013.

see [1]. Some take a machine learning approach by extracting features from sequences to predict a functional assignment. Profile methods take advantage of patterns such as conserved regions or structural qualities found in the proteins themselves and compare them to annotated proteins.

Network models use biological network data to predict protein function. Primitive network models determine the function of an individual protein based upon the known function of proteins in its immediate neighborhood. Graph theoretic models model the entire network simultaneously, and diffuse information between the proteins according to the edges defined in the graph. Probabilistic network models also model the entire network, but assign labels with an associated probability.

Markov Random Fields (MRFs), one kind of probabilistic network model, have often been used in predicting protein functions from network data [2,3,4]. Deng *et. al.* [2] defined an MRF model for predicting protein functional annotations, and laid the basic framework. They define an MRF over a protein-protein interaction (PPI) network, where pairwise interactions between proteins are modeled by factors in the MRF. Kourmpetis *et. al.* [3] extended this method by improving parameter estimation through multiple parameter estimation steps. These MRF models mostly use protein-protein interaction data, but there are many other biological network sources that can suggest functional similarity such as genetic interaction networks. Here, we show that the use of additional network data, integrated with a conditional random field (CRF) model, can give increased performance over the previous methods.

Perhaps the most similar approach to ours used MRFs and included GI networks [5,6]. These methods have some limitations; their models do not make use of continuous data, assume independence between network sources, and use a single parameter estimation step. A follow-up paper uses a more sophisticated parameter estimation scheme [7]. In this paper, we take into account many more sources of evidence in a CRF model which corrects these flaws.

2 Method

Previous models using MRFs need to either model the relationships between the input data, which can become complicated, and is essentially unnecessary, or assume independence [5,6,7], which is often wrong. Conditional random fields are the discriminative version of MRFs which model the dependence of the output on the input rather than the full joint distribution of the input and output. Our contribution to the field - to extend the previous framework in [2] and [3] to a CRF model with multiple sources of data - is described here.

2.1 Conditional Random Fields

A conditional random field (CRF) is a discriminative graphical model which splits the variables into two sets; input variables X, and output variables Y. We are not interested in modeling the relationships between variables within

X; these may not be related to the problem we wish to solve, and can even be very difficult to model. By conditioning over X, we assume a dependence upon X, but make no assumptions upon the distributions of variables within X. This allows us to model more complex relationships between the variables in Y and X. A CRF is represented as a factor graph, in which the random variables are represented as nodes, and factors describe the dependencies between them. The CRF distribution is defined in terms of its factors $f \in F$, conditioned over X:

$$p(Y|\theta, X) = \frac{1}{Z(X)} \prod_{f \in F} \psi_f(Y_f, X) \tag{1}$$

$$Z(\theta, X) = \sum_{y'_1} ... \sum_{y'_n} \prod_{f \in F} \psi_f(Y'_f, X) \ , \tag{2}$$

where Z is a normalization function dependent upon X, and Y_f are all the variables in Y involved in factor f.

2.2 The Model

We therefore, for the problem of protein function prediction, represent the protein labels and sources of evidence between them (e.g. physical interactions, co-expression) as nodes in a factor graph, where factors describe the relationships between them. The CRF model has to integrate multiple sources of evidence which each describe in some form the functional relationships between proteins. We consider one function at a time. For each protein p_i, we introduce a variable y_i, which describes its label (1 if the protein has the function, 0 otherwise). Networks which describe interactions between proteins define their context with respect to their functional labels. An edge from network σ which describes an interaction between proteins p_i and p_j, is represented by the random variable $ev_\sigma(i.j)$. We group all ev variables in the set X. Figure 1 shows the basic outline of the method.

The probability of a particular labeling for the graph is defined in terms of the factorization of the model. Figure 2 illustrates how a simple 5-protein network can be factorized with two kinds of factor nodes, pairwise factors, ψ_p and singular factors ψ_s. These are defined in terms of their respective energy functions, U_p and U_s:

$$\psi_p(y_i, y_j; \theta, X) = \exp\{U_p(y_i, y_j; \theta, X)\}$$
$$\psi_s(y_i; \theta, X) = \exp\{U_s(y_i; \theta, X)\} \ .$$

U_s is defined as α if the node has the label, and 0 otherwise:

$$U_s(y_i; \theta, X) = \alpha y_i \ , \tag{3}$$

where α is a parameter. U_p is a function which depends on whether (a) both nodes have the label, (b) one node has the label, or (c) neither node has the

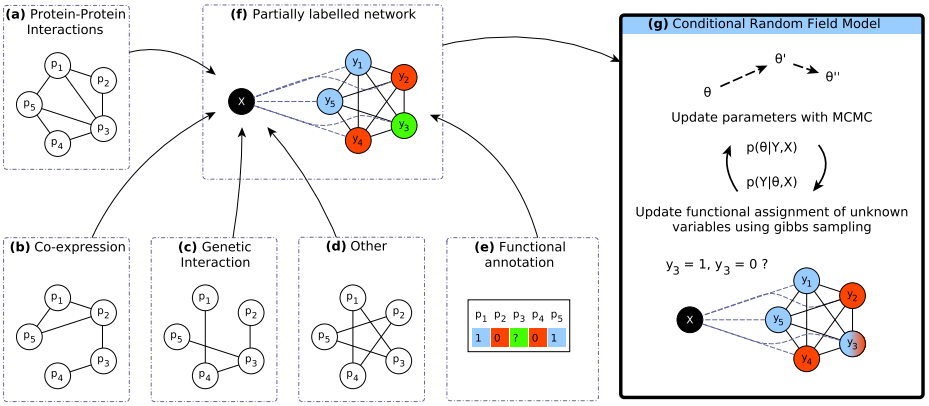


Fig. 1. Conditional random field analysis for protein function prediction. **a-d**: Different sources of evidence between proteins define the functional relationships between proteins. **e**: Variables that describe the functional annotations of proteins are introduced. **f**: Dependencies between the variables are described by the graphical model. **g**: Missing labels are inferred in an iterative scheme.

label. It connects two protein label variables, and the evidence of interactions between them. For a single source of evidence σ this can be expressed as::

$$\begin{aligned} U_\sigma(y_i, y_j; \theta, X) = & \beta_{\sigma,11} y_i y_j ev_\sigma(i,j) + & \text{(a)} \\ & \beta_{\sigma,10}[(1-y_i)y_j + y_i(1-y_j)] ev_\sigma(i,j) + & \text{(b)} \\ & \beta_{\sigma,00}(1-y_i)(1-y_j) ev_\sigma(i,j)\ , & \text{(c)} \end{aligned} \qquad (4)$$

where $(\beta_{\sigma,11}, \beta_{\sigma,01}, \beta_{\sigma,00})$ are parameters. All sources of evidence are combined to form the general pairwise energy function U_p:

$$U_p(y_i, y_j; \theta, X) = \sum_\sigma U_\sigma(y_i, y_j; \theta, X)\ . \qquad (5)$$

When there is only one source of evidence, the model is equivalent to that of [3]. The structure of the label network is no longer explicitly defined by any single biological network, so in order to ensure that nodes are able to use all relationships available to them in X, the graph is by default fully connected between all the variables $y_1 \cdots y_n$; Each pair $(y_i, y_j) \in Y \times Y$ are connected by a pairwise factor. In the event where there is no evidence between two proteins, the relevant potential becomes equal to 1, and therefore does not influence the statistical distribution (1) ($\exp\{0\} = 1$).

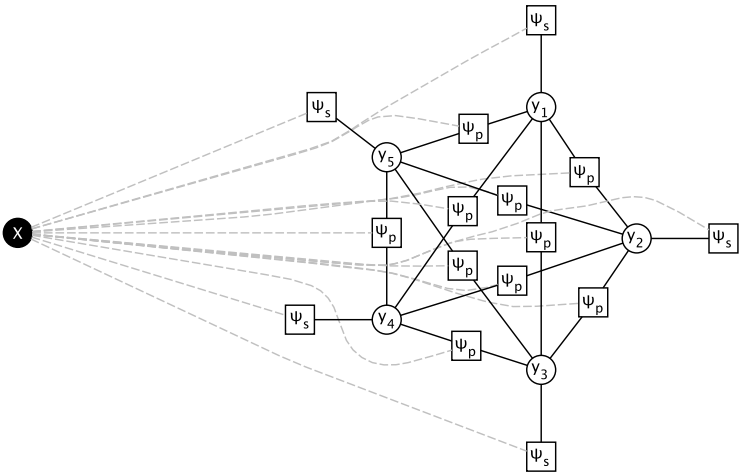


Fig. 2. CRF factorization of the model: Each factor ψ_s and ψ_p in the graph is dependent upon the additional data X

2.3 Conditional Probability

We can express the conditional probability of an individual node being positive in terms of the logistic function, as in [2]:

$$p(y_i = 1|Y_{-i}, \theta, X) = \frac{\psi_s(y_i) \prod_{(y_i,y_j)\in Y\times Y} \psi_p(y_i, y_j, X)}{\sum_{y'_i} \psi_s(y'_i) \prod_{(y'_i,y'_j)\in Y\times Y} \psi_p(y'_i, y'_j, X)} , \tag{6}$$

$$= \frac{\exp\{\ell\}}{1+\exp\{\ell\}} , \tag{7}$$

where ℓ in this case is defined as the log-odds of the probability of the labels at the node, and Y_{-i} refers to all nodes in Y with the exception of i.

$$\ell = \log \frac{p(y_i = 1|Y_{-i}, \theta, X)}{1 - p(y_i = 1|Y_{-i}, \theta, X)} \tag{8}$$

$$= \alpha + \sum_{\sigma} \sum_{(y_i,y_j)\in Y\times Y} [\delta_\sigma y_j ev_\sigma(i,j) + \epsilon_\sigma(1-y_j) ev_\sigma(i,j)] .$$

For each data source σ, two parameters, δ_σ and ϵ_σ are introduced, which replace the β parameters. The derivation is given in [8].

2.4 Inference

We wish to maximize (1), which is normally done with belief propagation. However, because of the size of the network and the number of factors, this can be

intractable. Instead, a Gibbs sampling algorithm is used to predict the labels. The Gibbs sampler operates by sampling a new label for each node from the conditional distribution at that node (7), and using the updated labels to sample new labels for the remaining proteins. The new labels will be used in the following iteration.

2.5 Parameter Estimation

For general graphs as these, exact parameter estimation is intractable [9]. Instead, parameters $\theta = (\alpha, \delta_\sigma, \epsilon_\sigma)$ are found by maximizing the pseudo-likelihood function (PLF), which has been described as a good approximation to the likelihood function. Kourmpetis [3] improved the parameter estimation by re-estimating them iteratively. This measure assumes that the density factorizes into the conditional distributions at each node:

$$PLF(Y|\theta, X) = \prod_{y_i \in Y} p(y_i|Y_{-i}, \theta, X) \; . \tag{9}$$

Each conditional (7) reminds us of the logistic function; logistic regression is thus used to update the parameters θ.

At each iteration, we re-estimate the parameters using the entire set of proteins, including the unknown ones for which new labels were just predicted. The new parameters θ^* are accepted over the previous parameters θ with a Metropolis step, (i.e., with probability):

$$A(\theta^*, \theta) = \min\left(1, \frac{PLF(Y|\theta^*, X)}{PLF(Y|\theta, X)}\right) \; , \tag{10}$$

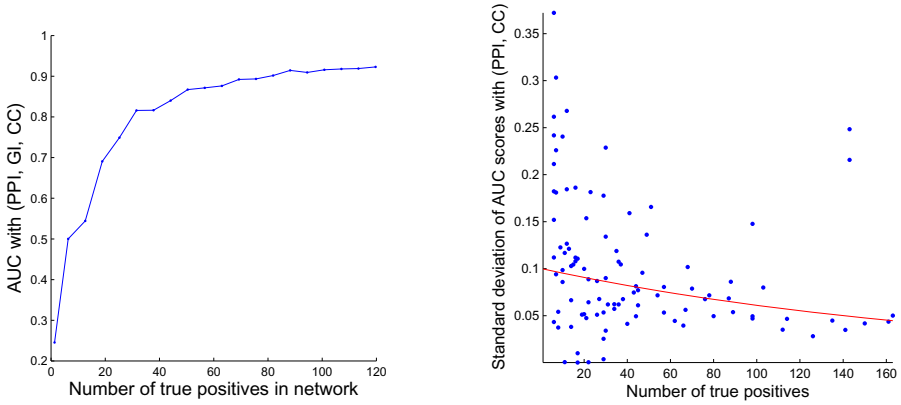
i.e. the new parameters will be accepted with probability $A(\theta^*, \theta)$ from a standard uniform distribution.

3 Experimental Setup

The yeast (*Saccharomyces cerevisiae*) genome is well studied and comes with a plethora of data, making it an excellent organism to test on. In earlier work [3], a PPI dataset from [10] was used. In order to gain additional information sources, different datasets were used. Functional annotations for yeast were taken from the Gene Ontology website. This gave functional annotations to 6383 proteins for 4631 GO terms [11]. An outline of the initialization of the algorithm is given in [8].

3.1 Dataset

We collected a dataset of the following sources:



(a) (b)

Fig. 3. Performance depends on the class distribution in the training set: **a**: The learning curve when using co-citation and genetic interaction data. **b**: As the number of true positives increases, we observe a lower standard deviation in AUC scores. Here co-citation data is used as additional data. The line indicates a best fit for an exponential decay function, with $y = 0.1 \exp(-0.005x)$.

(**PPI**) Protein-Protein Interaction: Collected from BioGRID [12].
(**KI**) Kinase Interaction: Collected from PhosphoGRID [13].
(**GI**) Genetic Interaction: Collected from BioGRID.
(**CX**) Co-expression: Collected from MegaYeast [14].
(**CC**) Co-citation: Collected from STRINGdb [15].

The PPI and KI networks were combined into the same network. The co-citation scores from STRINGdb was mapped onto a logistic curve between 0 and 1. (It is not clear how STRINGdb calculated the original co-citation scores).

3.2 Performance Evaluation

100 functions were randomly selected from the *Biological Processes* and *Molecular Function* ontologies in the Gene Ontology (GO). For each GO term, we select a test set of 300 proteins to mark as unknown; this constitutes the testing set in a cross validation procedure. To ensure that there is positive data in the test set, it is constructed such that it contains approximately 20% of all positive labels for that particular GO term. For each function, the model was trained 10 times using different test sets, and the Area Under the Receiver Operator Characteristic Curve (AUC) score is calculated. For more information, see [8].

4 Results and Discussion

Before discussing any results on the data, we report how the model behaves in training and predicting. i.e. whether parameter estimates converge, and predictions are reliable.

4.1 Model Behavior

Iterative parameter estimation improves on the initial prediction performance. Parameter estimates converge very quickly, usually within one or two steps (data not shown); In the remaining iterations the values only oscillate a little, as in [3]. Despite having good parameter estimates, we cannot stop iterating after just a few steps due to the changing labels after each Gibbs step; The Gibbs sampler needs time to build up a good average of the label probabilities. Like in [3], we observe that the intercept parameter, α is estimated well in the first step (Deng *et. al.*'s estimate), while the δ_σ and ϵ_σ parameters usually are not. Since the intercept parameter is less sensitive to the individual labelings of the nodes we can imagine that it is easier to estimate.

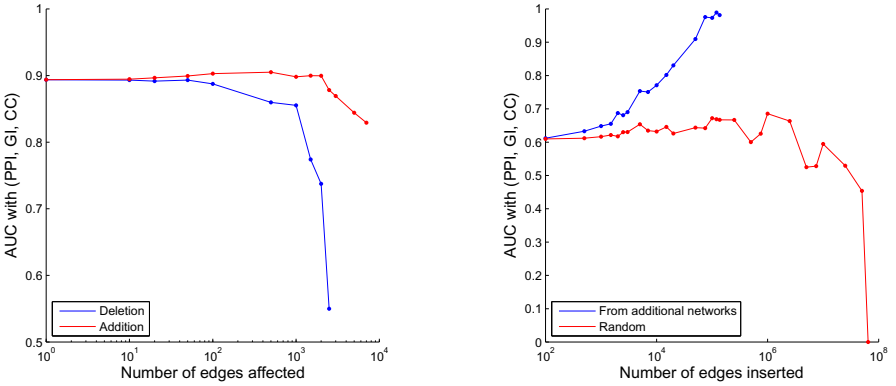
Performance depends on class balance in the training set. The number of proteins annotated with a function (the number of true positives) influences the performance of the method. By removing functional annotations from a function which has many true positives[1], we can see how the performance improves as the number of true positives increases. Figure 3a shows a learning curve which illustrates that as we add more true positives, we are able to predict more accurately.

Illustrated in figure 3b is how the standard deviation of predictions for each function varies depending upon the number of true positives. With more true positives (i.e. more training data), we have a lower standard deviation. An exponential decay function is fit to the data to demonstrate the trend of the data; The standard deviation actually decreases. Unfortunately, a function rarely has that many true positives [8].

The precision of predictions with our method is comparable to that of [3]. This is important to us; if using more data were to give us (on average) a better prediction but with a large variance, it would be useless in practice. Functions for which the performance is bad, generally have a high standard deviation [8].

The model is robust to noise. Rather surprisingly, the model is quite insensitive to random noise. We test the model by independently deleting and adding edges to the relevant 'unknown' proteins. These tests were run under optimal conditions; A specific function[1] was selected for which there were many true positives (many proteins have been annotated with it), and for which the model already performs well. Figure 4a indicates that the model is more sensitive to edge deletions, however, this is due to the fact that there are a limited number of edges that can be deleted before none are left.

[1] *GO:0004672* 'Protamine kinase activity', from the molecular function ontology.



(a) (b)

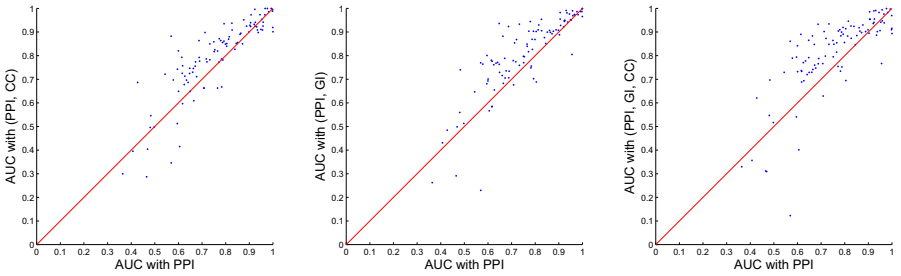
Fig. 4. Model robustness: **a**: The resilience of the method when deleting edges from, or adding edges into the network. **b**: When adding random edges to the network, we do not get a significant performance increase.

A little noise acts as a regularization. Adding random edges can provide a kind of regularization, possibly by adding edges smoothing the network (i.e. making neighborhoods more similar to each other), making the classification task easier. Evident from figure 4a and studied in figure 4b, is that adding a few random edges gives a slightly better performance. For a given function[1], we create two distinct subnetworks from all the sources of evidence, (a) containing only edges also present in the PPI network and (b) all other edges. We compare adding edges to (a) either randomly, or from (b). Adding useful edges from (b) improves performance drastically, in contrast, adding random edges helps only a little, until the graph becomes saturated with edges and no relationships are distinguishable anymore.

4.2 Some Additional Sources Improve Prediction

Adding data sources gives better performance in some cases, and this is reflected in Figures 5a-c. These figures plot the performance of the baseline ([3], using PPI data), against the performance of our method. Any point above the diagonal line means that that particular function has a better performance.

On average, CC data gives us a large improvement over the previous methods, GI data a slightly larger improvement, and the combination of the two an even larger improvement. Figure 5c shows the results from combination of CC and GI networks, which gives the largest improvement on average. The increase in performance is due to the additional data, which provides new information on relationships between proteins. Note the fact that CC data may be good is because of possible bias (two proteins may be co-cited because they *already*



(a) (b) (c)

Fig. 5. The effect of using additional data: We compare the performance per function against that of [3]. On the x-axis are the AUC scores of the algorithm per function, using only protein-protein interaction data (equivalent to [3]), and on the y-axis the AUC scores of the algorithm using additional data. **a**: Using CC data. **b**: Using GI data. **c**: Using CC *a*nd GI data.

have the same functional annotation). CX data invariably has a detrimental effect upon performance [8].

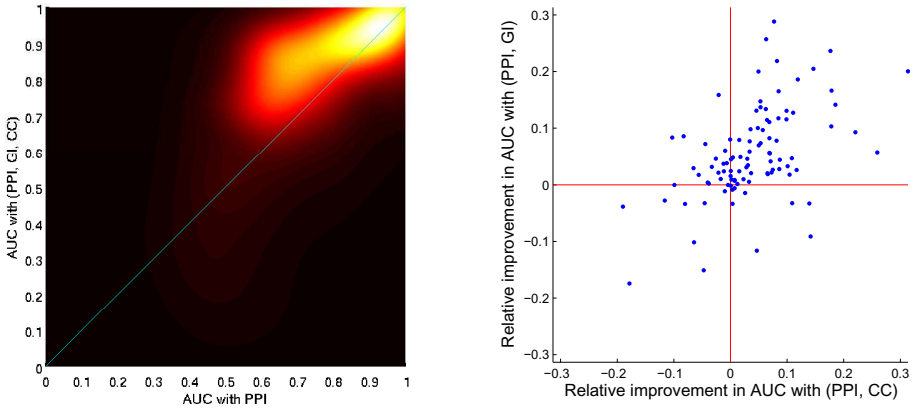
When there are very few true positives for a function, the additional parameters make it difficult to train the model properly, and performance may suffer. In such cases it may be advisable to revert to a single source of evidence.

When we consider each performance pair (only PPI, additional sources) the center of a normal distribution and take into account their standard deviations, we can sum the distributions to get an idea of the performance over all functions. This plot is seen in figure 6a. Most of the density is above the diagonal (over 85%), indicating that the predictions with our method are expected to be better than those in [3].

Data sources complement each other. Figure 6b plots the improvement relative to the Kourmpetis *et. al.* model when using different data sources. It shows that even though there is often a common improvement, correlation is not very high. A function for which one data source helps does not necessarily benefit from another data source. This means that different sources of evidence supply information valuable to predict different functions; The sources of evidence complement each other and are not interchangeable.

5 Discussion

Here we present, for the first time, a CRF model that can be used to easily and effectively predict protein function. The ability to accurately predict which proteins are involved in a function is of great importance to biologists. Whereas MRF models have been used before, our CRF model demonstrates that additional information helps improve prediction. GI and CC networks provide the most useful information. Data sources which have a continuous value would be



(a) (b)

Fig. 6. **a**: A density plot reveals the general trend of model performance. **b**: Different sources of evidence supply different information to the model. We calculate the improvement in prediction but taking the difference in AUC scores when running the model with additional data, and when running it without. They are not highly correlated ($\rho_{GI,CC} \approx 0.57$).

able to describe more subtle relationships between proteins, rather than just strong ones, but such sources are hard to find.

We describe thoroughly the construction of the model and analyze its performance. A more complex factorization could be constructed to describe more complicated relationships between proteins. Furthermore, a nonlinear combination of sources of evidence could give rise to a richer description of the requirements for functional similarity (e.g. proteins should interact *and* be co-expressed).

Despite any improvements, this method is still stochastic, as evident in the variance experienced over multiple runs. Consequently, in practice the model would have to be run multiple times to ascertain exactly which proteins are consistently the most highly ranked.

Acknowledgments. We thank Marc Hulsman for providing some valuable insights and the data sources.

References

1. Radivojac, P., Clark, W.T., Oron, T.R., Schnoes, A.M., Wittkop, T., Sokolov, A., Graim, K., Funk, E.A.: A large-scale evaluation of computational protein function prediction. Nature Methods 10(3) (January 2013)
2. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F.: Prediction of protein function using protein-protein interaction data. Journal of Computational Biology 10(6), 947–960 (2003)

3. Kourmpetis, Y.A.I., van Dijk, A.D.J., Bink, M.C.A.M., van Ham, R.C.H.J., ter Braak, C.J.F.: Bayesian Markov random field analysis for protein function prediction based on network data. PLoS ONE 5(2), 9293 (2010)
4. Letovsky, S., Kasif, S.: Predicting protein function from protein/protein interaction data: a probabilistic approach. Bioinformatics 19(suppl. 1), 197–204 (2003)
5. Deng, M., Chen, T., Sun, F.: An integrated probabilistic model for functional prediction of proteins. Journal of Computational Biology 11(2-3), 463–475 (2004)
6. Deng, M., Tu, Z., Sun, F., Chen, T.: Mapping Gene Ontology to proteins based on protein-protein interaction data. Bioinformatics 20(6), 895–902 (2004)
7. Kourmpetis, Y.A.I., van Dijk, A.D.J., van Ham, R.C.H.J., ter Braak, C.J.F.: Genome-wide computational function prediction of *Arabidopsis* proteins by integration of multiple data sources. Plant Physiology 155(1), 271–281 (2011)
8. Gehrmann, T.: Conditional random fields for protein function prediction. M.sc. thesis, Delft University of Technology, Delft (2012)
9. Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields (November 2010)
10. Collins, S.R., Kemmeren, P., Zhao, X.C., Greenblatt, J.F., Spencer, F., Holstege, F.C.P., Weissman, J.S., Krogan, N.J.: Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. Molecular & Cellular Proteomics 6(3), 439–450 (2007)
11. Michael Ashburner, C.A.: Creating the gene ontology resource: design and implementation. Genome Research 11(8), 1425–1433 (2001)
12. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic Acids Research 34(suppl. 1), D535–D539 (2006)
13. Stark, C., Su, T.C., Breitkreutz, A., Lourenco, P., Dahabieh, M., Breitkreutz, B.J., Tyers, M., Sadowski, I.: PhosphoGRID: a database of experimentally verified in vivo protein phosphorylation sites from the budding yeast Saccharomyces cerevisiae. Database 2010 (January 2010)
14. Gasch, A.: Megayeast expression dataset (August 2012), `http://gasch.genetics.wisc.edu/datasets.html`
15. Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L.J., von Mering, C.: The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Research 39(database issue), D561–D568 (2011)

Enhancing Protein Fold Prediction Accuracy Using Evolutionary and Structural Features

Abdollah Dehzangi[1,2], Kuldip Paliwal[1], James Lyons[1], Alok Sharma[3], and Abdul Sattar[1,2]

[1] Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Brisbane, Australia
[2] National ICT Australia (NICTA), Brisbane, Australia
[3] University of the South Pacific, Fiji
{a.dehzangi,k.paliwal,j.lyons,a.sattar}@griffith.edu.au, sharma_al@usp.ac.fj

Abstract. Protein fold recognition (PFR) is considered as an important step towards the protein structure prediction problem. It also provides crucial information about the functionality of the proteins. Despite all the efforts that have been made during the past two decades, finding an accurate and fast computational approach to solve PFR still remains a challenging problem for bioinformatics and computational biology. It has been shown that extracting features which contain significant local and global discriminatory information plays a key role in addressing this problem. In this study, we propose the concept of segmented-based feature extraction technique to provide local evolutionary information embedded in Position Specific Scoring Matrix (PSSM) and structural information embedded in the predicted secondary structure of proteins using SPINE-X. We also employ the concept of occurrence feature to extract global discriminatory information from PSSM and SPINE-X. By applying a Support Vector Machine (SVM) to our extracted features, we enhance the protein fold prediction accuracy to 7.4% over the best results reported in the literature.

Keywords: Protein Fold Recognition, Feature Extraction, Segmented distribution, Segmented Auto Covariance, Occurrence, Support Vector Machine (SVM).

1 Introduction

Protein Fold Recognition (PFR) is defined as assigning a given protein to a fold (among a finite number of folds) that represents its functionality as well as its major tertiary structure. Therefore, PFR is considered as an important step towards the protein structure prediction problem. Despite all the efforts that have been made so far to find an effective computational approach to solve this problem, it still remains an unsolved problem for computational biology. From the pattern recognition perspective, PFR is defined as solving a multi-class classification task. Therefore, extracting features that capture significant

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 196–207, 2013.

global and local discriminatory information as well as the classification technique being used play the main roles in solving this problem. During the past two decades, a wide range of classification techniques have been used for PFR [1–9]. Among the classifiers employed to tackle this problem, *Support Vector Machine (SVM)* based classifiers have attained the best results [10, 11]. However, the most significant enhancement of PFR accuracy has been achieved by relying on the feature extraction approaches rather than the classification techniques being used [1, 9, 10, 12–14]. In most of the studies that addressed PFR by feature extraction techniques, global discriminatory information has been represented using the composition of the amino acids feature group (the occurrence of the amino acids along the protein sequence divided by the length of protein sequence [1, 8]). However, it has been shown that this feature group is not able to adequately reveal global information [15]. Furthermore, composition feature group is not able to capture information regarding the length of the protein sequence that was shown as an effective feature for PFR [13].

Compared to the methods adopted to extract global discriminatory information, a wider range of methods were used to extract local discriminatory information for PFR such as, pseudo amino acid composition [3, 8, 9], cross covariance [10], auto covariance [10], bi-gram [11, 14], and tri-gram [16]. Despite the significant local discriminatory information provided using these approaches, most of these methods produce large number of features which makes them computationally expensive for large protein data banks (e.g. cross covariance and tri-gram [10, 16]). At the same time, in all these methods the whole protein sequence as a single entity have been used to extract local information. In another words, they aimed to extract local information by exploring whole protein sequence as a global entity. Therefore, they could not appropriately explore local information embedded in protein sequence. Furthermore, despite all the efforts have been made to enhance the protein fold prediction accuracy so far, its prediction accuracy remains limited especially when the sequential similarity rate is low.

In this study, we aim at enhancing protein fold prediction accuracy by addressing these limitations. We propose segmented-base feature extraction to extract local evolutionary information embedded in *Position Specific Scoring Matrix (PSSM)* as well as structural information embedded in the predicted secondary structure using SPINE-X. We also employ the concept of an occurrence feature of the transformed protein sequence using evolutionary and structural information embedded in PSSM and SPINE-X to extract adequate global discriminatory information for PFR. By applying SVM to our extracted features we enhance the protein fold prediction accuracy to 7.4% better than the highest reported results found in the literature.

2 Data Sets

In this study, two data sets namely TG and EDD are used to investigate the performance of our proposed methods. The TG data set introduced by [15] consists of 1612 proteins belonging to 30 folds with less than 25% sequential

similarities. TG is extracted from *Structural Classification of Proteins (SCOP)* 1.73 which has been previously used to investigate the performance of proposed methods for PFR when the sequential similarity is very low [13, 15, 17]. We also extract EDD (extended version of DD data set [1] which is extracted from SCOP 1.75). This data set consists of 3418 proteins belonging to 27 folds that was used originally in DD data set with less than 40% sequential similarities. The EDD data set extracted from an older version of SCOP has been widely used for PFR [5, 10, 11]. Using this data set enables us to directly compare our results with previously reported results found in the literature.

3 Feature Extraction Method

In this study, we rely on PSSM and the predicted secondary structure using SPINE-X to extract evolutionary and structural information respectively. PSSM is calculated by applying PSIBLAST [18] to EDD and TG data sets (using NCBI's non redundant (NR) database with its cut off value (E) set to 0.001). PSSM consists of an $L \times 20$ matrix (L is the length of a protein and the columns of the matrices represent 20 amino acids). It provides the substitution probability of a given amino acid based on its position along a protein sequence.

We also use predicted secondary structure using SPINE-X which was recently proposed by [19] and attained better results (especially for the coded area) than PSIPRED on predicting protein secondary structure [20]. Given a protein sequence, it returns an $L \times 3$ matrix (which will be referred to as SPINE-M for the rest of this study) consisting of the normalized probability of contribution of a given amino acid based on its position along the protein sequence to build one of the three secondary structure elements namely, α-helix, β-strands, and coils. It also returns a transformed version of the protein sequence (also extracted from SPINE-M) in which each amino acid along the protein sequence is replaced with H (represents helix), E (represents strand), or C (represents coil) based on its tendency to incorporate in building one of these secondary structure elements. In this study, we will refer to this sequence as the structural consensus sequence. It is expected that predicted secondary structure using SPINE-X provides significant structural information for PFR similar to or even better than PSIPRED due to its better performance [19]. In continuation, the global and local features extracted in this study will be explained in detail.

3.1 Global Features

To extract global discriminatory information embedded in PSSM and SPINE-M we mainly relied on the concept of the occurrence feature. We extract evolutionary and structural consensus sequence-based occurrence from the transformed protein sequence using PSSM and SPINE-M respectively. We also extract semi-occurrence feature group directly from PSSM and SPINE-M which represents the summation of the substitution probability of the amino acids and normalized probability of secondary structure elements respectively.

Consensus Sequence-Based Occurrence: In this method, we extract occurrence of the amino acids as well as occurrence of the secondary structure elements derived from the evolutionary-based and the structural-based consensus sequences respectively. To extract the occurrence feature group from the evolutionary consensus sequence, we first need to extract this sequence from PSSM. In the evolutionary consensus sequence, amino acids along the original protein sequence $(O_1, O_2, ..., O_L)$ are replaced with the corresponding amino acids with the maximum substitution probability $(C_1, C_2, ..., C_L)$. This is done in the following two steps. In the first step, for a given amino acid, the index of the amino acid with the highest substitution probability is calculated as follows:

$$I_i = \text{argmax}\{P_{ij} : 1 \leq j \leq 20\},\ 1 \leq i \leq L, \tag{1}$$

where P_{ij} is the substitution probability of the amino acid at location i with the j^{th} amino acid in PSSM. In the second step, we replace the amino acid at i^{th} location of original protein sequence by the $I_{i^{th}}$ amino acid to form the consensus sequence. After calculating the evolutionary consensus sequence, we count the occurrence of each amino acid (for all the 20 amino acids) along this sequence and produce the occurrence feature from the evolutionary based consensus sequence which we call (AAO). Similarly, we calculate the occurrence of each *secondary structure elements (SSEO)* (for all three elements) in the structural consensus sequence and extract the corresponding feature group. The occurrence feature group is used in this study as the global descriptor of the proteins since it maintains the information regarding the length of protein sequence which is disregarded using composition feature group [2, 5].

Semi-Occurrence: In this method, we calculate semi-occurrence feature group from both PSSM and SPINE-M. It is called semi-occurrence because instead of using the protein sequence directly to calculate the occurrence of each amino acid, we calculate the summation of the substitution probability for each amino acid from the PSSM or normalized frequency of each secondary structure element from SPINE-M. The semi-occurrence derived from the PSSM (*PSSM_AAO*) is calculated as follows:

$$\text{PSSM-AAO}_j = \sum_{i=1}^{L} P_{ij}, (j = 1, ..., 20). \tag{2}$$

In a similar manner, we calculate the semi-occurrence of the normalized frequency of the secondary structure elements from SPINE-M (*SPINE_SSEO*) as follows:

$$\text{SPINE-SSEO}_j = \sum_{i=1}^{L} S_{ij}, (j = 1, 2, 3), \tag{3}$$

where S_{ij} is the normalized probability of the occurrence of the j^{th} secondary structure element for the i^{th} amino acid in the SPINE-M. These feature groups

are able to provide important global discriminatory information about the substitution probability of the amino acids as well as normalized frequency of secondary structure elements based on PSSM and SPINE-M. For the rest of this study, the combination of all these four global feature groups (AAO + SSEO + PSSM-AAO + SPINE-SSEO) will be referred as F_{global} (consisting of 46 features in total).

3.2 Local Features

To extract these features, we extract distribution and auto covariance features using segmentation method. In this manner, we are able to provide more local information compared to use of whole protein sequence as a global entity to extract these features.

Segmented Distribution: This method is specifically proposed to extract more local discriminatory information for PFR based on the amino acids' substitution probability with each other (extracted from PSSM) as well as their tendency to incorporate in one of the secondary structure elements (extracted from SPINE-M). For PSSM, for the j^{th} column, we first calculate the total sum of substitution probability $T_j = \sum_{i=1}^{L} P_{ij}$. Then, starting from the first row of PSSM (which corresponds to the first amino acid in the protein sequence) we sum the substitution probabilities corresponding to the j^{th} column until reaching to less than or equal to F_P (segmentation factor) of T_j ($S_1 = \sum_{i=1}^{I_j^1} P_{ij}$). I_j^1 is the number of amino acids such that the summation of their substitution probability is equal to S_1 and is the corresponding feature for this segment. We calculate I_j^2 by summing the substitution probability of amino acids (again, starting from the first row of PSSM) until reaching $2 \times F_P$ of T_j. Similarly, I_j^2 is the number of amino acids such that the summation of their substitution probability is equal to S_2 ($2 \times F_P$ of T_j) and is the corresponding feature for this segment. In this study F_P is set to 25% since it attained similar performance as adopting 10% and 5% for this parameter. In other words, dividing the protein sequence into four segments provide similar local discriminatory information in comparison with dividing it to 10 or 20.

We also calculate I_j^3, I_j^4 features for the j^{th} column of PSSM. Dissimilar to I_j^1 and I_j^2, we start from the last row of PSSM (corresponding to the last amino acids of the protein sequence). To calculate I_j^3, starting from the last row of PSSM, we sum the substitution probabilities of amino acids until reaching less than or equal to F_P of T_j. In the similar manner, we calculate I_j^4, summing substitution probability of amino acids (starting from the last row of PSSM) until reaching to $2 \times F_P$ of total sum (T_j). In this manner, we also cover whole protein sequence as well (50% of T_j is covered by starting from the first row and 50% of T_j is covered by starting from the last row). Therefore, for a given column in PSSM we calculate 4 segmented distribution features (which in total $4 \times 20 = 80$ features are extracted corresponding to 20 columns in PSSM) to build segmented distribution feature group (called *PSSM_SD*).

In a similar manner, we calculate the segmented distribution feature group of the normalized frequency of the secondary structure elements from SPINE-M (called *SPINE_SD*) using $F_S = 25\%$ (where F_s is used as the distribution factor for SPINE-M equivalent to F_P used for PSSM) and respectively extract $3 \times 4 = 12$ features in total for all three elements.

Segmented Auto Covariance: The concept of auto covariance have been widely used in the literature to capture local discriminatory information and has attained better results compared to similar methods used for this task such as bi-gram [11, 14] or tri-gram features [16]. Pseudo amino acid composition based features are good examples of these types of features [3, 21]. These features have been computed using the whole protein sequence as a single entity for feature extraction. Therefore, they could not adequately explore the local discriminatory information embedded in protein sequence [10]. In the present study, we extend the concept of segmented distribution features as described in the previous subsection to compute the auto covariance features. This provides more local evolutionary and structural information from PSSM and SPINE-M. First for PSSM, we segment the protein sequence using $F_P = 25\%$. Using a procedure similar to the one described in the previous subsection, for the j^{th} column in PSSM we divide the protein sequence into 4 segments (from first amino acid corresponding to first row of PSSM until reaching I_j^1; from first amino acid corresponding to first row of PSSM until reaching I_j^2; from last amino acid corresponding to the last row of PSSM until reaching I_j^3; and from last amino acid corresponding to the last row of PSSM until reaching I_j^4). we calculate auto covariance feature using K_P(distance factor used for PSSM for each segment) as follows:

$$\text{PSSM-seg}_{n,m,j} = \frac{1}{(I_j^n - m)} \sum_{i=1}^{I_j^n - m} (P_{i,j} - P_{ave,j}) \times (P_{(i+m),j} - P_{ave,j}),$$
$$(n = 1, 2, 3, 4 \ \& \ m = 1, ..., K_P \ \& \ j = 1, ..., 20), \quad (4)$$

where, $P_{ave,j}$ is the average substitution probability for the j^{th} column in PSSM. We also compute the global auto covariance coefficient (K_P features) as follows:

$$\text{PSSM-AC}_{m,j} = \frac{1}{(L - m)} \sum_{i=1}^{L-m} (P_{i,j} - P_{ave,j}) \times (P_{(i+m),j} - P_{ave,j}),$$
$$(m = 1, ..., K_P \ \& \ j = 1, ..., 20). \quad (5)$$

Thus, we extract a total of ($2K_P + 2K_P + K_P = 5K_P$) auto covariance features ($2K_P$ features for segments corresponding to I_j^1 and I_j^2, $2K_P$ features for segments corresponding to I_j^3 and I_j^4 and K_P features corresponding to global auto covariance) in this manner. Then by combining PSSM-AC and PSSM-seg (extracted for all 20 columns of PSSM) we build the corresponding feature group which is called PSSM-SAC ($20 \times (5 \times K_P)$) features in total).

This procedure is also repeated for SPINE-M in the same way (K_S is used as the distance factor for SPINE-M equivalent to K_P used for PSSM) for all three columns of SPINE-M and segmented auto covariance of normalized frequency of secondary structure elements are extracted as follows:

$$\text{SPINE-seg}_{n,m,j} = \frac{1}{(I^n_{max} - m)} \sum_{i=1}^{I^n_{max}-m} (S_{i,j} - S_{ave,j}) \times (S_{(i+m),j} - S_{ave,j}),$$
$$(n = 1, 2, 3, 4 \ \& \ m = 1, ..., K_S \ \& \ j = 1, 2, 3), \quad (6)$$

where, $S_{ave,j}$ is the average substitution probability for the j^{th} column in SPINE-M. Similarly, the global auto covariance is computed as follows:

$$\text{SPINE-AC}_{m,j} = \frac{1}{(L - m)} \sum_{i=1}^{L-m} (S_{i,j} - S_{ave,j}) \times (S_{(i+m),j} - S_{ave,j}),$$
$$(m = 1, ..., K_S \ \& \ j = 1, 2, 3). \quad (7)$$

The combination of SPINE-seg and SPINE-AC builds SPINE-SAC consisting of $3 \times (5K_S)$ features in total (extracted for all three columns of SPINE-M).

4 Support Vector Machine

In pattern recognition, SVM is considered as the-state-of-the-art classification technique. It was introduced by [22] aiming at finding the *Maximum Margin Hyper-plane (MMH)* based on the concept of support vector theory to minimize classification error. It transforms the input data to higher dimensionality using the kernel function to find support vectors. The classification of some known points in input space $\mathbf{x}_i$ is y_i which is defined to be either -1 or +1. If x' is a point in input space with unknown classification then:

$$y' = sign\left(\sum_{i=1}^{n} a_i y_i K(\mathbf{x}_i, \mathbf{x}') + b \right), \quad (8)$$

where y' is the predicted class of point $\mathbf{x}'$. The function *K()* is the kernel function; n is the number of support vectors and a_i are adjustable weights and b is the bias. The best results reported in the literature for PFR was attained using this classifier [4, 10, 11, 16]. In this study, the SVM classifier implemented in LIBSVM (C-SVC type) toolbox with *Radial Basis Function (RBF)* as its kernel function is used [23]. RBF kernel is adopted here due to its better performance than other kernels functions (e.g. polynomial kernel, linear kernel, and sigmoid [10]). In this study, the width parameter γ in addition to the cost parameter C of the SVM are optimized using grid search algorithm implemented in the LIBSVM package.

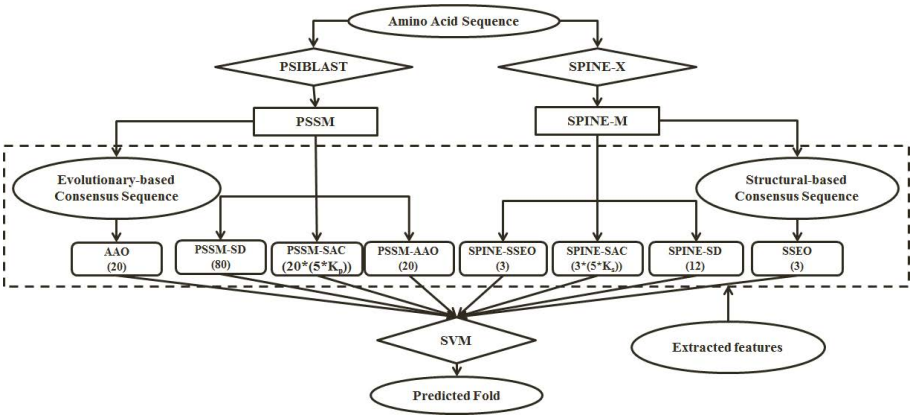


Fig. 1. The general architecture of our proposed feature extraction model. The number of features extracted in each feature group is shown in the brackets below the feature groups' names.

5 Results and Discussion

We construct the input feature vector to use with SVM consisting of our extracted feature (F_{global} + PSSM-SD + SPINE-SD + PSSM-SAC + SPINE-SAC). The architecture of our proposed system is shown in Figure 1. To evaluate the performance of our proposed methods, 10-fold cross validation evaluation criterion is adopted in this study as it was often used for this task in the literature [1, 5, 11, 15]. We first investigate the impact of our proposed method for PFR with respect to the K_p and K_s parameters in PSSM-SAC and SPINE-SAC respectively. Then we investigate the impact of each of the proposed feature groups in this study separately on the achieved prediction accuracy. Finally, we compare our achieved results with previously reported results for the PFR.

5.1 Investigating the Impact of K_p and K_s

As it was mentioned earlier, K_p and K_s values between 1 and 10 are investigated here (since it was shown in [10] that using a distance factor larger than 10 to extract auto covariance feature group attains similar results with using 10 for PFR). To do this, in 10 different experiments, we apply SVM to our proposed feature vector while K_p and K_s are monotonically increased from 1 to 10 ($K_p = 1$ and $K_s = 1$, $K_p = 2$ and $K_s = 2$, ... , $K_p = 10$ and $K_s = 10$). The results for this experiment is shown in Figure 2. We also calculate the SVM parameters on EDD data set (where $K_p = 10$ and $K_s = 10$) for our proposed feature vector using the grid search algorithm. Calculated parameters are used for the rest of this study (to avoid over tuning parameters) for both TG and EDD data sets (where C = 0.07 and $\gamma = 100$). Note that the TG data sets have not been used at all for parameter tuning.

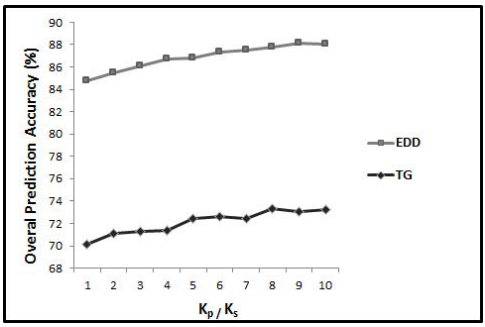
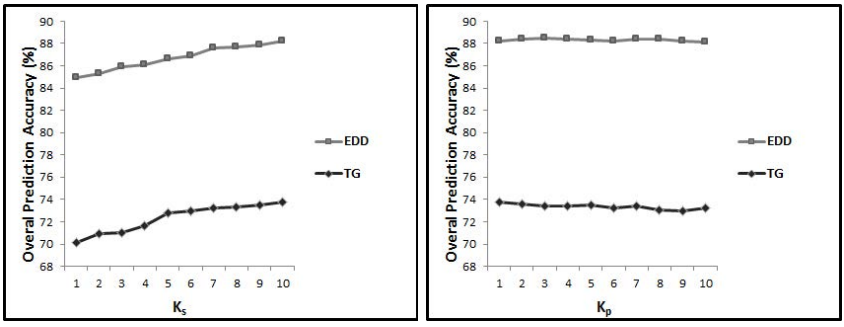


Fig. 2. The results achieved for TG and EDD data sets with respect to K_p and K_s which are monotonically increase from 1 to 10



(a) The impact of increasing K_s from 1 to 10 while $K_p = 1$ for EDD and TG data sets

(b) The impact of increasing K_p from 1 to 10 while $K_s = 10$ for EDD and TG data sets

Fig. 3. Investigating the effective values for K_s and K_p in our proposed feature extraction method

As we can see, increasing the K_p and K_s, prediction accuracy almost monotonically increases as well. Using $K_p = 10$ and $K_s = 10$, we reach 88.1% and 73.1% prediction accuracies for EDD and TG data sets respectively. However, it is not clear which one of K_p and K_s has the main impact on the achieved results. To investigate the effectiveness of K_p and K_s, two different experiments are conducted on the EDD data set. First, we set the value of $K_p = 1$ and in 10 different experiments, increase the value of K_s from 1 to 10 (Figure 3.a). As we can see, increasing K_s monotonically increases the prediction accuracy and setting $K_s = 10$ attain the best result for this task. In a different experiment, we set the value of $K_s = 10$ and in 10 different experiments, increase the value of K_p from 1 to 10. As we can see in Figure 3.b, the performance does not change by increasing the K_p. As it is shown in Figure 3.a and 3.b, similar results are achieved for the TG data set. In other words, using segmented auto covariance approach, we are able to reveal more local discriminatory information from PSSM and SPINE-M based on the concept of auto covariance compared to previous studies ($K_P = 1$ and $K_S = 10$). It is dramatically lower than the

number of features used in [10] and [11] to reveal this information. Therefore, for the rest of this study K_p and K_s are set to 1 and 10 respectively.

5.2 Determining the Effect of the Proposed Feature Groups on the Protein Fold Prediction Accuracy

In continuation, we investigate the effectiveness of each of the feature groups used in this study separately to our reported protein fold prediction accuracy. The results are shown in Table 1. As we can see, all the feature groups used to reveal global and local discriminatory information are effectively contribute to the achieved protein fold prediction enhancement.

Table 1. The impact of proposed feature groups proposed in this study (using SVM classifier) to enhance protein structural class prediction accuracy (in %). For PSSM-SAC and SPINE-SAC, the values of K_p and K_s are respectively set to 1 and 10.

Combination of features	EDD	TG
F_{global}	74.7	58.7
F_{global} + PSSM-SD	79.4	62.6
F_{global} + SPINE-SD	79.1	63.6
F_{global} + PSSM-SD + SPINE-SD	82.3	66.7
F_{global} + PSSM-SAC	80.1	64.0
F_{global} + SPINE-SAC	84.1	68.2
F_{global} + PSSM-SAC + SPINE-SAC	86.1	71.8
F_{global} + PSSM-SD + SPINE-SD + PSSM-SAC	87.5	72.6
F_{global} + PSSM-SD + SPINE-SD + SPINE-SAC	87.1	72.8
PSSM-SD + SPINE-SD + PSSM-SAC + SPINE-SAC	85.9	71.1
F_{global} + PSSM-SD + SPINE-SD + PSSM-SAC + SPINE-SAC	88.2	73.8

5.3 Comparison with the Existing Methods

We compared the results achieved by applying SVM to the combination of features proposed in this study (F_{global}, PSSM-SAC, PSSM-SD, SPINE-SAC, SPINE-SD where K_p and K_s are set to 1 and 10 respectively) which will be referred as PSSM-SPINE-S (388 features in total) with the best results reported in the literature. The results are shown in Table 2. As we can see, we report up to 73.8% and 88.2% prediction accuracies for TG and EDD data sets respectively. These results are up to 7.4% and 2.3% better than the highest reported results for these two data sets that are achieved by reproducing the results reported in [10] for TG and EDD data sets respectively. The enhancement achieved compared to other similar approaches to reveal more local information such as bi-gram [11] and tri-gram [16] is much more significant (over 11% for EDD and TG data sets). The higher enhancement achieved for TG data set compared to [10] shows that our method is more effective when the sequential similarity rate is very low (up to 25%). It is also important to highlight that we outperformed [10] using 388 features compared to 4000 features used in that study. Therefore, our proposed methodology is able to significantly enhance protein fold prediction accuracy compared to the state-of-the-art methods found in the literature and at the same time reduce the number of features used for this task dramatically. In other words, we are able to provide more local and global information

Table 2. Comparison of the results reported EDD and TG data sets (in %). Note that column named No. is referring to the number of features.

Ref.	Features	No.	Method	EDD	TG
[15]	AAO (from original protein sequence)	20	LDA	46.9	36.3
[15]	AAC (from original protein sequence)	20	LDA	40.9	32.0
[1]	Physicochemical Features + AAC	125	SVM	50.1	39.5
[13]	Physicochemical Features + AAC	220	ANN(RBF)	52.8	41.9
[17]	Threading	-	Naive Bayes	70.3	55.3
[16]	PF (bi-gram)	400	SVM	75.2	52.7
[16]	TF (Tri-gram)	8000	SVM	71.0	49.4
[11]	Combination of bi-gram features	2400	SVM	69.9	55.0
[5]	PSIPRED and PSSM features	242	SVM	77.5	60.1
[10]	ACCfold-AC	200	SVM	80.1	58.8
[10]	ACCfold-ACC	4000	SVM	85.9	66.4
This study	PSSM-SPINE-S	388	SVM	88.2	73.8

from PSSM and SPINE-X for PFR compared to previously proposed approaches found in the literature.

6 Conclusion

In this study, we have proposed two novel segmentation based feature extraction techniques to reveal more local discriminatory information embedded in PSSM and SPINE-X. We also employed the concept of occurrence feature group and extend it to provide more global discriminatory information from PSSM and SPINE-X for PFR compared to previously used methods for this task. Then by applying SVM to the combination of our features extracted we significantly enhanced protein fold prediction accuracy compared to previously reported results in the literature. We achieved up to 73.8% and 88.2% prediction accuracies, up to 7.4% and 2.3% better than the highest results reported for TG and EDD data sets respectively [10]. These enhancements were achieved by using less than 1/10 of features used previously in [10]. In other words, we were able to extract more potential local and global discriminatory information for PFR compared to previously proposed methods found in the literature using fewer features.

References

1. Ding, C., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17, 349–358 (2001)
2. Chen, K., Kurgan, L.A.: Pfres: protein fold classification by using evolutionary information and predicted secondary structure. Bioinformatics 23(21), 2843–2850 (2007)
3. Shen, H.B., Chou, K.C.: Ensemble classifier for protein fold pattern recognition. Bioinformatics 22, 1717–1722 (2006)
4. Damoulas, T., Girolami, M.: Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection. Bioinformatics 24(10), 1264–1270 (2008)
5. Deschavanne, P., Tuffery, P.: Enhanced protein fold recognition using a structural alphabet. Proteins: Structure, Function, and Bioinformatics 76(1), 129–137 (2009)

6. Dehzangi, A., Phon-Amnuaisuk, S., Dehzangi, O.: Using random forest for protein fold prediction problem: An empirical study. Journal of Information Science and Engineering 26(6), 1941–1956 (2010)
7. Dehzangi, A., Phon-Amnuaisuk, S., Dehzangi, O.: Enhancing protein fold prediction accuracy by using ensemble of different classifiers. Australian Journal of Intelligent Information Processing Systems 26(4), 32–40 (2010)
8. Kavousi, K., Sadeghi, M., Moshiri, B., Araabi, B.N., Moosavi-Movahedi, A.A.: Evidence theoretic protein fold classification based on the concept of hyperfold. Mathematical Biosciences 240(2), 148–160 (2012)
9. Yang, T., Kecman, V., Cao, L., Zhang, C., Huang, J.Z.: Margin-based ensemble classifier for protein fold recognition. Expert Systems with Applications 38, 12348–12355 (2011)
10. Dong, Q., Zhou, S., Guan, G.: A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. Bioinformatics 25(20), 2655–2662 (2009)
11. Shamim, M.T.A., Anwaruddin, M., Nagarajaram, H.A.: Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. Bioinformatics 23(24), 3320–3327 (2007)
12. Chen, K., Stach, W., Homaeian, L., Kurgan, L.: ifc2: an integrated web-server for improved prediction of protein structural class, fold type, and secondary structure content. Amino Acids 40, 963–973 (2011)
13. Dehzangi, A., Phon-Amnuaisuk, S.: Fold prediction problem: The application of new physical and physicochemical- based features. Protein and Peptide Letters 18(2), 174–185 (2011)
14. Sharma, A., Lyons, J., Dehzangi, A., Paliwal, K.K.: A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. Journal of Theoretical Biology 320(0), 41–46 (2013)
15. Taguchi, Y.H., Gromiha, M.M.: Application of amino acid occurrence for discriminating different folding types of globular proteins. BMC Bioinformatics 8(1) (2007)
16. Ghanty, P., Pal, N.R.: Prediction of protein folds: Extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers. IEEE Transactions on NanoBioscience 8(1), 100–110 (2009)
17. Gromiha, M.M.: Multiple contact network is a key determinant to protein folding rates. Journal of Chemical Information and Modeling 49(4), 1130–1135 (2009)
18. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J.H., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Research 17, 3389–3402 (1997)
19. Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., Zhou, Y.: Spine x: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. Journal of Computational Chemistry 33(3), 259–267 (2012)
20. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. Journal of Molecular Biology 292(2), 195–202 (1999)
21. Shen, H.B., Chou, K.C.: Predicting protein fold pattern with functional domain and sequential evolution information. Journal of Theoretical Biology 256(3), 441–446 (2009)
22. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc. (1995)
23. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines (2001)

Exploring Potential Discriminatory Information Embedded in PSSM to Enhance Protein Structural Class Prediction Accuracy

Abdollah Dehzangi[1,2], Kuldip Paliwal[1], James Lyons[1], Alok Sharma[3], and Abdul Sattar[1,2]

[1] Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Brisbane, Australia
[2] National ICT Australia (NICTA), Brisbane, Australia
[3] University of the South Pacific, Fiji
{a.dehzangi,k.paliwal,j.lyons,a.sattar}@griffith.edu.au, sharma_al@usp.ac.fj

Abstract. Determining the structural class of a given protein can provide important information about its functionality and its general tertiary structure. In the last two decades, the protein structural class prediction problem has attracted tremendous attention and its prediction accuracy has been significantly improved. Features extracted from the *Position Specific Scoring Matrix (PSSM)* have played an important role to achieve this enhancement. However, this information has not been adequately explored since the protein structural class prediction accuracy relying on PSSM for feature extraction still remains limited. In this study, to explore this potential, we propose segmentation-based feature extraction technique based on the concepts of amino acids' distribution and auto covariance. By applying a *Support Vector Machine (SVM)* to our extracted features, we enhance protein structural class prediction accuracy up to 16% over similar studies found in the literature. We achieve over 90% and 80% prediction accuracies for 25PDB and 1189 benchmarks respectively by solely relying on the PSSM for feature extraction.

Keywords: Protein Structural Class Prediction Problem, Feature Extraction, Segmented distribution, Segmented Auto Covariance, Support Vector Machine (SVM).

1 Introduction

Protein structural class prediction problem is defined as assigning a given protein to one of four structural classes namely all-α, all-β, $\alpha + \beta$, and α/β [1]. Protein structural class prediction can provide important information about the functionality of proteins as well as their general tertiary structure. Despite all the efforts that have been made to find a fast computational approach to solve this problem, especially for low homologous protein sequences, it still remains unsolved for computational biology and bioinformatics [2–4].

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 208–219, 2013.

During the last two decades, a wide range of classification techniques have been proposed to tackle the protein structural class prediction problem such as, *Support Vector Machine (SVM)* [5–8], *Artificial Neural Network (ANN)* [9, 10], *Meta Classifiers* [11, 12], and ensembles of classifiers [13–15]. Among the proposed classification techniques used to tackle this problem, SVM has attained the best results [7, 16–18]. Similarly, a wide range of features have been proposed and used to reveal more discriminatory information for this task [5, 16, 19]. More significant improvement for protein structural class prediction accuracy has come from the new features being introduced rather than the classification technique being used for this task [16, 17, 20].

The first group of features that significantly enhanced the protein structural class prediction accuracy were extracted from the evolutionary information embedded in the *Position Specific Scoring Matrix (PSSM)* [21]. Latter on, several feature extraction techniques were proposed to explore the potential local and global discriminatory information embedded in PSSM to tackle this problem such as composition of the amino acids [8], pseudo amino acid composition [2], dipeptide composition [8], and auto covariance [17]. However, the discriminatory information embedded in PSSM has not been adequately explored since the prediction accuracy relying on these features remains limited. Further enhancement for the protein structural class prediction accuracy has been achieved by relying on the structural information extracted [7, 16] from the predicted secondary structure of proteins using PSIPRED [22]. despite a wide range of feature extraction techniques being explored [5, 7, 8, 20], the protein structural class prediction accuracy relying on structural information has not been improved adequately since the study of Mizianty and Kurgan in 2009 [16]. This highlights the need for novel feature extraction techniques relying on the alternative sources for feature extraction.

In this study, we propose two segmented feature extraction techniques based on the concepts of distribution and auto covariance methods to explore local discriminatory information embedded in the PSSM. We also use the concept of occurrence of the amino acids to explore global discriminatory information embedded in PSSM rather than composition of the amino acids that has been widely used for this task to capture the information regarding the length of the protein sequence [16, 17]. By applying SVM to our extracted features we achieve over 90% and 80% protein structural class prediction accuracies for 25PDB and 1189 benchmarks respectively. We enhance the protein structural class prediction accuracy for up to 16% compared to smilar studies which have used PSSM for feature extraction.

2 Benchmarks

In this study, two popular benchmarks that have been widely used for the protein structural class prediction problem namely, 25PDB and 1189 benchmarks are used. The 25PDB benchmark was introduced in [19] consists of 1673 proteins with less than 25% sequential similarities (the homology range between 22%

and 45%). This benchmark was extracted from 25% PDBSELECTED which includes high resolution protein sequences in the *Protein Data Bank (PDB)* [23]. Therefore, this benchmark is considered as a reliable representative of proteins in the twilight zone (proteins with the sequence similarities between 20% to 45%). Hence, this benchmark is employed in this study as the main source to investigate the performance of our proposed techniques.

The 1189 benchmark is a popular benchmark that has been widely used in the literature. This benchmark was introduced by [3] consisted of 1189 proteins. However, 97 proteins were dropped from this benchmark in later studies [19] to address further correction of *Structural Classification of Proteins (SCOP)* [24]. As the result, current version of this benchmark consists of 1092 proteins with less than 40% sequential similarities. Dissimilar to 25PDB, this benchmark includes proteins with low resolutions as well. Therefore, despite higher sequential similarity among proteins in this benchmark, lower prediction accuracies have been reported in the literature for this benchmark compared to 25PDB using similar approaches [5, 7, 8]. This benchmark is mainly used in this study to compare our results directly with previously reported results as well as tuning the classification and feature extraction parameters while 25PDB benchmark is not used at all in the tuning step.

3 Feature Extraction Method

Since our proposed features are all extracted directly from PSSM, we need to first produce this matrix. To calculate PSSM, PSI-BLAST [21] is applied for both 25PDB and 1189 benchmarks (using NCBI's non redundant (NR) database while its cut off value (E) is set to 0.001). PSSM provides the substitution probability of a given amino acid based on its position in a protein sequence with all 20 amino acids. It consists of two L $\times$ 20 matrices (where L is the length of protein sequence and 20 columns are representatives of 20 amino acids). The first matrix provides the log-odds of the amino acids substitution probabilities and it is called PSSM_cons while the second matrix provides normalized substitution probability and it is called PSSM_probs. Since PSSM_cons has been widely used in the literature for feature extraction [16, 17], it is also adopted in this study.

To explore potential local and global discriminatory information embedded in PSSM, four feature groups are proposed and used in this study. These feature groups are, consensus sequence-based occurrence of the amino acids (AAO), semi occurrence of the amino acids (PSSM-AAO), segmented distribution (PSSM-SD), and segmented auto covariance (PSSM-SAC). The first two feature groups are proposed to reveal global discriminatory information while the remaining two methods are proposed to reveal local discriminatory information embedded in PSSM. These four feature extraction methods are explained in detail in the following subsections.

3.1 Consensus Sequence-Based Occurrence (AAO)

To extract global discriminatory information embedded in PSSM, we first extract the occurrence of the amino acids feature group from the consensus sequence derived from PSSM. In the protein consensus sequence, amino acids along the original protein sequence $(O_1, O_2, ..., O_L)$ are replaced with the corresponding amino acids with the maximum substitution probabilities in PSSM $(C_1, C_2, ..., C_L)$. This is done in the following two steps. In the first step, the index of the amino acid with the highest substitution probability (based on its position in the protein sequence) is calculated as follows:

$$I_i = \text{argmax}\{P_{ij} : 1 \leq j \leq 20\},\ 1 \leq i \leq L, \tag{1}$$

where P_{ij} is the substitution probability of the amino acid at location i with the j^{th} amino acid in PSSM_cons. In the second step, we replace the amino acid at i^{th} location of original protein sequence by the $I_{i^{th}}$ amino acid to form the consensus sequence. After calculating the consensus sequence, we count the number of occurrence of each amino acid (for all 20 amino acids) along the consensus sequence and return the corresponding values. Therefore, a feature group consisting of 20 features is calculated. The occurrence feature group as the global descriptor of the proteins is used in this study since it maintains the information regarding the length of protein sequence which is discarded using the composition feature group (occurrence of amino acids divided by the length of the protein sequence (AAC) [16]).

3.2 Semi Occurrence (PSSM-AAO)

This feature group is directly extracted from the PSSM. It is called semi occurrence because it is not calculated in the similar manner to the occurrence feature group as it was explained in previous subsection. Instead, it is produced by summation of the substitution score of a given amino acid with all the amino acids along the protein sequence which is calculated as follows:

$$PSSM\text{-}AAO_j = \sum_{i=1}^{L} P_{ij}\ ,\ (j = 1, ..., 20). \tag{2}$$

This feature group is able to provide important global discriminatory information about the substitution probability of the amino acids [17]. Different to composition of the amino acid extracted from PSSM (which is called PSSM-AAC in [17]), PSSM-AAO maintains the information regarding to the length of protein sequence. In PSSM-AAC the the summation of substitution probabilities of the amino acids are divided by the length of protein sequence.

3.3 Segmented Distribution (PSSM-SD)

This method is specifically proposed to add more local discriminatory information about how the amino acids, based on their substitution probability with each

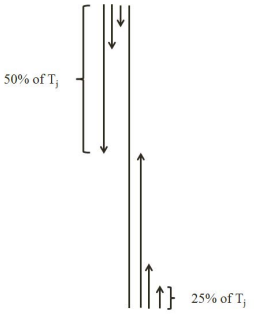


Fig. 1. The segmentation method used to extract PSSM-SD feature group

other (extracted from PSSM), are distributed along the protein sequence. We propose this segmentation method in the manner where segments of a protein sequence are of unequal lengths and each segment is represented by a distribution feature which is computed as follows. First, for the j^{th} column in the PSSM, we calculate the total substitution probability $T_j = \sum_{i=1}^{L} P_{ij}$. Then, starting from the first row of PSSM, we calculate the partial sum S_1 of the substitution probabilities of the first i amino acids until reaching to *25%* of the total sum $S_1 = \sum_{i=1}^{I_j^1} P_{ij}$. Using the distribution factor $F = 25\%$, we calculate the I_j^1. The I_j^1 corresponds to the number of the amino acids such that the summation of their substitution probabilities is less than or equal to the $F = 25\%$ of (T_j). Similarly, we calculate the partial sum of the first i amino acids (starting from the first row of PSSM) until reaching $2 \times F = 50\%$ of the total sum $S_2 = \sum_{i=1}^{I_j^2} P_{ij}$ and calculate the I_j^2 corresponding to the number of amino acids such that the summation of their substitution probabilities is less than or equal to $F = 50\%$ of the total T_j.

We repeat the same process beginning from the last row of the PSSM for the j^{th} column. We calculate the partial sum of the substitution probability of the first i amino acids until reaching $F = 25\%$ and $2 \times F = 50\%$ of the total sum which are $S_3 = \sum_{i=1}^{I_j^3} P_{ij}$ and $S_4 = \sum_{i=1}^{I_j^4} P_{ij}$ respectively and calculate the I_j^3 and I_j^4. I_j^3 and I_j^4 correspond to the number of amino acids such that the summation of their substitution probability is less than or equal to F and $2 \times F$ of T_j respectively (starting from the last row of PSSM). In this manner we extract four segmented distribution features for each column in PSSM. The method used to calculate PSSM-SD is shown in Figure 1. We repeat the same process for all 20 columns corresponding to 20 amino acids in PSSM and extract 80 features in total in this feature group $(4 \times 20 = 80)$. Note that $F = 25\%$ is adopted in this study due it s better performance compared to use of $F = 10\%$ and $F = 5\%$ explored experimentally by the authors. In the other word, using four segments is sufficient for providing adequate local discriminatory information compared to the use of 10 or 20 segments.

3.4 Segmented Auto Covariance (PSSM-SAC)

The concept of auto covariance has been widely used in the literature to capture local discriminatory information and has attained better results compared to similar methods used for this task such as dipeptide composition [8, 17]. Pseudo amino acid composition based features are good examples of these types of features [2, 4]. These features have been computed using the whole protein sequence as a single entity for feature extraction. Therefore, they could not adequately explore the local sequence order information embedded in protein sequence [17]. In the present study, we extend the concept of segmented distribution features as described in the previous subsection to compute the auto covariance features from the segmented protein sequence. This is done to enforce local discriminatory information extracted from PSSM.

To extract this feature group, we calculate the auto covariance of the substitution probability of the amino acids using K as the distance factor for each segment of proteins generated using segmented distribution in the following manner. Starting from the first row of PSSM, for the j^{th} column of PSSM, we calculate K auto covariance features for the first I_j^1. Similarly, we calculate auto covariance for the first I_j^2 amino acids. Then starting from the last row of PSSM for the j^{th} column of PSSM, We repeat the same process for I_j^3, and I_j^4 (I_j^1, I_j^2, I_j^3, and I_j^4 are calculated from the previous subsection). This process is repeated for all 20 columns of PSSM and corresponding features are calculated as follows:

$$\text{PSSM-seg}_{n,m,j} = \frac{1}{(I_j^n - m)} \sum_{i=1}^{I_j^n - m} (P_{i,j} - P_{ave,j}) \times (P_{(i+m),j} - P_{ave,j}),$$
$$(n = 1, ..., 4 \ \& \ m = 1, ..., K \ \& \ j = 1, ..., 20), \quad (3)$$

where, $P_{ave,j}$ is the average substitution probability for the j^{th} column in PSSM. Note that $2 \times K$ auto covariance coefficients are computed in this manner by analyzing PSSM in the downward direction and $2 \times K$ auto covariance coefficients are computed in this manner by analyzing PSSM in the upward direction ($4 \times K$ features in total). We also compute the global auto covariance coefficient (K features) of PSSM as follows:

$$\text{PSSM-AC}_{m,j} = \frac{1}{(L - m)} \sum_{i=1}^{L-m} (P_{i,j} - P_{ave,j}) \times (P_{(i+m),j} - P_{ave,j}),$$
$$(m = 1, ..., K \ \& \ j = 1, ..., 20). \quad (4)$$

Thus, we have extracted a total of ($2K + 2K + K = 5K$) auto covariance features in this manner (for the j^{th} column of the PSSM). Therefore, for all 20 columns of the PSSM, segmented auto covariance of substitution probability of the amino acids are extracted and combined to build the corresponding feature group which will be referred to as PSSM-SAC (PSSM-seg + PSSM-AC which consists of $20 \times (5K)$) features in total).

4 Support Vector Machine

SVM was introduced by [25] to find the *Maximum Margin Hyper-plane (MMH)* based on the concept of the support vector theory to minimize classification error. It transforms the input data to higher dimension using the kernel function to be able to find support vectors (for nonlinear cases). The classification of some known points in input space $\mathbf{x}_i$ is y_i which is defined to be either -1 or +1. If x' is a point in input space with unknown classification then:

$$y' = sign\left(\sum_{i=1}^{n} a_i y_i K(\mathbf{x}_i, \mathbf{x}') + b\right), \tag{5}$$

where y' is the predicted class of point $\mathbf{x}'$. The function *K()* is the kernel function; n is the number of support vectors and a_i are adjustable weights and b is the bias. This classier is considered as the state-of-the-art classification techniques in the pattern recognition and attained the best results for the protein structural class prediction problem [7, 16, 17]. In this study, SVM classifier implemented in the LIBSVM (C-SVC type) toolbox using *Radial Basis Function (RBF)* as its kernel is used [26]. The γ in addition to the regularization parameter C (which also called the soft margin parameter) of the RBF kernel are optimized using grid search algorithm implemented in the LIBSVM package.

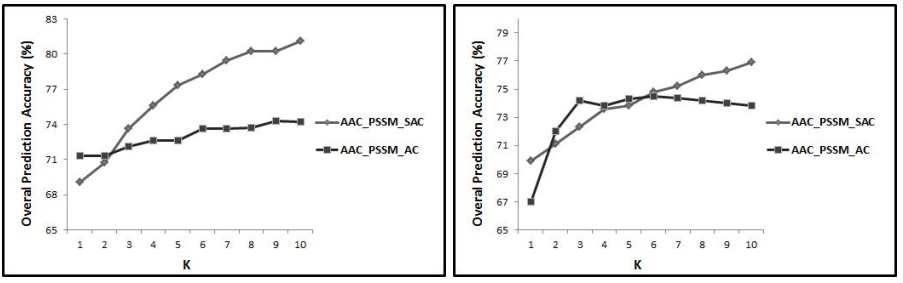
5 Results and Discussion

We first explore the effectiveness of the segmented auto covariance (PSSM-SAC) method compared to global auto covariance (PSSM-AC) used in [17]. PSSM-AC was used to explore local discriminatory information embedded in PSSM and attained the best results for this task. Then, one by one, we add the rest of the feature groups extracted in this study and explore their impact on the protein structural class prediction accuracy, separately. Finally, we compare the results reported in this study with the similar studies found in the literature for the protein structural class prediction problem. To evaluate the performance of our proposed methods and to be able to directly compare our results with previously studies, we adopt Jackknife cross validation as it was widely used for this task in the literature [16, 17, 19]

5.1 The Effectiveness of PSSM-SAC versus PSSM-AC

To investigate the effectiveness of PSSM-SAC compared to PSSM-AC we first reproduce the experiments conducted in [17]. In this experiment, PSSM-AC in combination of PSSM-AAC was used as the input feature group (called AAC-PSSM-AC) for different values of K (between 1 and 10) using an SVM classifier. We similarly combine the PSSM-SAC with PSSM-AAC (called AAC-PSSM-SAC) to be able to directly compare these two feature groups with respect to different values of distance factor K between 1 and 10 (using an SVM as it

was used in [17]). The results achieved for 25PDB and 1189 are respectively shown in Figure 2.a and Figure 2.b. As it is shown in these figures, increasing the K value, AAC-PSSM-SAC significantly outperform AAC-PSSM-AC. Using K = 10 we achieve up to 81.1% and 76.9% prediction accuracies respectively for 25PDB and 1189 benchmarks. This highlights the effectiveness of PSSM-SAC to extract local discriminatory information based on the concept of auto covariance from the PSSM. Note that our results using solely AAC-PSSM-SAC enhances the protein structural class prediction accuracy for up to 6% and 2.3% for 25PDB and 1189 benchmarks respectively compared to the best results found in the literature relying on PSSM for feature extraction. In continuation, we replaced PSSM-AAC with PSSM-AAO which enhances the protein structural class prediction accuracy for all 10 values of K between 0.5% and 2% (when increasing K from 1 to 10, the impact of AAO is reduced from almost 2% to 0.5%) which shows the effectiveness of using AAO compared to AAC. Therefore, for the rest of this study, AAO is used instead of AAC. We then use grid search algorithm on 1189 to optimize SVM parameters (C and γ) for AAO-PSSM-AC (where K = 10) to avoid over tuning. 25PDB also was not used at all for this task. The optimal values achieved for C and γ are respectively 500 and 0.05 which are used for the rest of this study.



(a) Comparison of the AAC_PSSM_AC and AAC_PSSM_SAC on 1189 benchmark

(b) Comparison of the AAC_PSSM_AC and AAC_PSSM_SAC on 25PDB benchmark

Fig. 2. Results achieved for AAC_PSSM_SAC and AAC_PSSM_AC with respect the value of K (Between 1 to 10) for 1189 and 25PDB benchmarks

5.2 The Effectiveness of PSSM-SD Feature Group

In continuation, we add the PSSM-SD feature group to the combination of PSSM-SAC and PSSM-AAO (AAO-PSSM-SAC) and study its impact for different values of K (between 1 and 10). The results achieved for 25PDB and 1189 benchmarks are shown in Figure 3. As we can see, by adding PSSM-SD, dissimilar to AAC-PSSM-SAC by increasing the value of K to 10, the prediction accuracy does not improve (it even slightly reduces). Therefore, adding PSSM-SD reduce the dependency to the value of K in PSSM-SAC to provide local

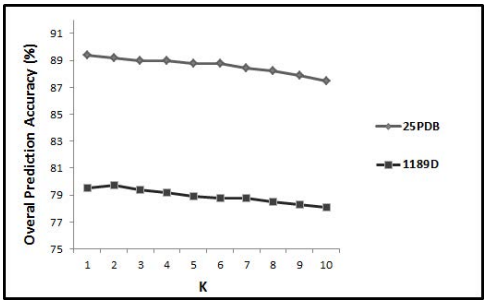


Fig. 3. The results achieved for combination of PSSM-AAO, PSSM-SAC, and PSSM-SD using SVM for different values of K (between 1 to 10) for 1189 and 25PDB benchmarks

information. In another word, we are able to increase the provided local information using PSSM-SD feature group and at the same time reduce the number of features. Using the combination of PSSM-AAO, PSSM-SAC, and PSSM-SD where K = 1 (20 + 100 + 80 = 200 features in total), we achieve up to 89.4% and 79.5% prediction accuracies for 25PDB and 1189 benchmarks respectively which are 15.3% and 4.9% better than the highest results reported for these benchmarks in the literature using features extracted from PSSM.

5.3 The Effectiveness of AAO Feature Group

In this Step, we add the AAO feature group to the combination of PSSM-AAO, PSSM-SAC (where K = 1), and PSSM-SD (20 + 20 + 100 + 80 = 220 features in total). By adding this feature group and applying SVM to these combination, we achieve up to 90.1% and 80.2% prediction accuracies respectively for 25PDB and 1189. These results are up to 16% and 5.6% respectively better than the best results reported for these two benchmarks using PSSM for feature extraction. It is important to highlight that these results are achieved using the same number of features used in [17] to achieve their best results for these two benchmark using PSSM for feature extraction. The results adding each feature group in each step is shown in Table.1. Note that in this table the impact of PSSM-SAC where K = 1 is shown while as it was explained in previous section, depend on the combination of feature groups being used, this impact has changed.

5.4 Performance Comparison with Existing Methods

In this section, the overall protein structural class prediction accuracy as well as prediction accuracy achieved for each structural class achieved by using the combination of our feature groups (PSSM-AAO + PSSM-SAC + PSSM-SD + AAO which will be referred as PSSM-S for simplicity) compared to previously reported results for this task are shown in Table 2 and Table 3. As we can see, we

Table 1. The impact of proposed feature extraction groups proposed in this study to enhance protein structural class prediction accuracy (in %)

Combination of features	Classifier	25PDB	1189
PSSM-AAO	SVM	65.5	62.4
PSSM-AAO + PSSM-SAC (K = 1)	SVM	69.9	69.1
PSSM-AAO + PSSM-SD	SVM	87.1	76.4
PSSM-AAO + PSSM-SAC (K = 1) + PSSM-SD	SVM	89.4	79.5
PSSM-AAO + PSSM-SAC (K = 1) + PSSM-SD + AAO	SVM	90.1	80.2
PSSM-AAO + PSSM-AC (K = 6) + PSSM-SD + AAO	SVM	89.1	78.1

Table 2. Comparison of the results reported for the 25PDB benchmark (in percentage %)

References	Method	All-α	All-β	α / β	α + β	Overall
[19]	Logistic Regression	69.1	61.6	60.1	38.3	57.1
[27]	Specific Tri-peptides	60.6	60.7	67.9	44.3	58.6
[13]	LLSC-PRED	75.2	67.5	62.1	44.0	62.2
[13]	SVM	77.4	66.4	61.3	45.4	62.7
[14]	SSA	92.6	83.7	80.5	65.9	81.5
[28]	SCPRED	92.6	80.1	74.0	71.0	62.7
[29]	CWT-PCA-SVM	76.5	67.3	66.8	45.8	64.0
[18]	AATP	81.9	74.7	75.1	55.8	71.7
[8]	AADP-PSSM	83.3	78.1	76.3	54.4	72.9
[17]	AAC-PSSM-AC	85.3	81.7	73.7	55.3	74.1
This Study	PSSM-S	93.8	92.8	92.6	81.7	90.1

Table 3. Comparison of the results reported for the 1189 benchmark (in percentage %)

References	Method	All-α	All-β	α / β	α + β	Overall
[3]	Bayes Classifier	54.8	57.1	75.2	22.2	53.8
[19]	Logistic Regression	57.0	62.9	64.7	25.3	53.9
[30]	FKNN	48.9	59.5	81.7	26.6	56.9
[27]	Specific Tri-peptides	-	-	-	-	59.9
[15]	IB1	65.3	67.7	79.9	40.7	64.7
[31]	SVM	75.8	75.2	82.6	31.8	67.6
[18]	AATP	72.7	85.4	82.9	42.7	72.6
[8]	AADP-PSSM	69.1	83.7	85.6	35.7	70.7
[17]	AAC-PSSM-AC	80.7	86.4	81.4	45.2	74.6
This Study	PSSM-S	93.3	85.1	77.6	65.6	80.2

not only significantly enhance the overall protein structural class prediction accuracy but also in most of the cases achieve better results for different structural classes. Relying solely on PSSM for feature extraction, we achieve over 90% and 80% prediction accuracies for 25PDB and 1189 benchmarks. It is important to highlight that we also achieved significantly higher results for 25PDB compared to studies which have used PSIPRED for feature extraction as well while it was relatively comparable for 1189 [7, 16].

6 Conclusion and Future Works

In this study, we proposed novel feature extraction methods to explore potential local and global discriminatory information embedded in PSSM for protein

structural class prediction problem. We proposed the concepts of segmented auto covariance and segmented distribution to extract this local information. We also employed the concept of occurrence to extract potential global discriminatory information directly from PSSM as well as the transformed protein sequence using PSSM. By applying SVM we showed the effectiveness of our proposed feature groups by enhancing protein structural class prediction accuracy for up to 16% and 5.6% for 25PDB and 1189 benchmarks respectively. We, for the first time, achieved over 90% and 80% (90.1% and 80.2%) protein structural class prediction accuracies for 25PDB and 1189 benchmarks respectively using PSSM for feature extraction. For our future work, we aim to study the effectiveness of structural information based on predicted secondary structure of proteins to enhance the protein structural class prediction accuracy, further.

References

1. Chothia, C.: The nature of the accessible and buried surfaces in proteins. Journal of Molecular Biology 105(1), 1–12 (1976)
2. Chou, K.C.: Progress in protein structural class prediction and its impact to bioinformatics and proteomics. Current Protein and Peptide Science 6, 423–436 (2005)
3. Wang, Z.X., Yuan, Z.: How good is prediction of protein structural class by the component-coupled method? Proteins: Structure, Function, and Bioinformatics 38(2), 165–175 (2000)
4. Chou, K.C.: Some remarks on protein attribute prediction and pseudo amino acid composition. Journal of Theoretical Biology 273(1), 236–247 (2011)
5. Yang, J.Y., Peng, Z.L., Chen, X.: Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. BMC Bioinformatics 11(suppl. 1), S9 (2010)
6. Li, Z.C., Zhou, X.B., Lin, Y.R., Zou, X.Y.: Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. Amino Acids 35(3), 581–590 (2008)
7. Zhang, S., Ding, S., Wang, T.: High-accuracy prediction of protein structural class for low-similarity sequences based on predicted secondary structure. Biochimie 93(4), 710–714 (2011)
8. Liu, T., Jia, C.: A high-accuracy protein structural class prediction algorithm using predicted secondary structural information. Journal of Theoretical Biology 267(3), 272–275 (2010)
9. Jahandideh, S., Abdolmaleki, P., Jahandideh, M., Asadabadi, E.B.: Novel two-stage hybrid neural discriminant model for predicting proteins structural classes. Biophysical Chemistry 128(1), 87–93 (2007)
10. Jahandideh, S., Abdolmaleki, P., Jahandideh, M., Hayatshahi, S.H.S.: Novel hybrid method for the evaluation of parameters contributing in determination of protein structural classes. Journal of Theoretical Biology 244(2), 275–281 (2007)
11. Cai, Y.D., Feng, K., Lu, W., Chou, K.: Using logitboost classifier to predict protein structural classes. Theoretical Biollogy 238, 172–176 (2006)
12. Jain, P., Hirst, J.: Automatic structure classification of small proteins using random forest. BMC Bioinformatics 11(1), 364 (2010)
13. Kurgan, L.A., Chen, K.: Prediction of protein structural class for the twilight zone sequences. Biochemical and Biophysical Research Communications 357(2), 453–460 (2007)

14. Kurgan, L.A., Zhang, T., Zhang, H., Shen, S., Ruan, J.: Secondary structure-based assignment of the protein structural classes. Amino Acids 35, 551–564 (2008)
15. Chen, K., Kurgan, L.A., Ruan, J.: Prediction of protein structural class using novel evolutionary collocation-based sequence representation. Journal of Computational Chemistry 29(10), 1596–1604 (2008)
16. Mizianty, M., Kurgan, L.A.: Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. BMC Bioinformatics 10(1), 414 (2009)
17. Liu, T., Geng, X., Zheng, X., Li, R., Wang, J.: Accurate prediction of protein structural class using auto covariance transformation of psi-blast profiles. Amino Acids 42, 2243–2249 (2012)
18. Zhang, S., Ye, F., Yuan, X.: Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via pssm. Journal of Biomolecular Structure and Dynamics 29(6), 1138–1146 (2012)
19. Kurgan, L.A., Homaeian, L.: Prediction of structural classes for protein sequences and domains - impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. Pattern Recognition 39, 2323–2343 (2006)
20. Yang, J.Y., Peng, Z.L., Yu, Z.G., Zhang, R.J., Anh, V., Wang, D.: Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. Journal of Theoretical Biology 257(4), 618–626 (2009)
21. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J.H., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Research 17, 3389–3402 (1997)
22. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. Journal of Molecular Biology 292(2), 195–202 (1999)
23. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Research 28(1), 235–242 (2000)
24. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: Scop: A structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 247(4), 536–540 (1995)
25. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc. (1995)
26. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines (2001)
27. Costantini, S., Facchiano, A.M.: Prediction of the protein structural class by specific peptide frequencies. Biochimie 91(2), 226–229 (2009)
28. Kurgan, L.A., Cios, K.J., Chen, K.: Scpred: Accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. BMC Bioinformatics 9, 226 (2008)
29. Li, Z.C., Zhou, X.B., Dai, Z., Zou, X.Y.: Prediction of protein structural classes by chou's pseudo amino acid composition: approached using continuous wavelet transform and principal component analysis. Amino Acids 37, 415–425 (2009)
30. Zhang, T.L., Ding, Y.S., Chou, K.C.: Prediction protein structural classes with pseudo amino acid composition: approximate entropy and hydrophobicity pattern. Theoretical Biology 250, 186–193 (2008)
31. Chen, C., Zhou, X., Tian, Y., Zou, X., Cai, P.: Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network. Analytical Biochemistry 357(1), 116–121 (2006)

Inferring the Association Network from p53 Sequence Alignment Using Granular Evaluations

David K.Y. Chiu and Ramya Manjunath

University of Guelph, Guelph, Ontario, Canada
{dchiu@uoguelph.ca}

Abstract. The relationship connecting the biomolecular sequence, the molecular structure, and the biological function is of extreme importance in nanostructure analysis such as drug discovery. Previous studies involving multiple sequence alignment of biomolecules have demonstrated that associated sites are indicative of the structural and functional characteristics of biomolecules, comparable to methods such as consensus sequences analysis. In this paper, a new method to detect associated sites in aligned sequence ensembles is proposed. It involves the use of multiple sub-tables (or levels) of two-dimensional contingency table analysis. The idea is to incorporate analysis by using a concept known as granular computing, which represents information at different levels of granularity. The analysis involves two phases. The first phase includes labeling of the molecular sites in the p53 protein multiple sequence alignment according to the detected associated patterns. The sites are consequently labeled into three different types based on their site characteristics: 1) conserved sites, 2) associated sites and 3) hypervariate sites. In the second phase, the significance of the extracted site patterns is evaluated with respect to targeted structural and functional characteristics of the p53 protein. The results indicate that the extracted site patterns are significantly associated with some of the known functionalities of p53, a cancer suppressor. Furthermore, when these sites are aligned with p63 and p73, the homologs of p53 without the same cancer suppressing property, based on the common domains, the sites significantly discriminate between the human sequences of the p53 family. Therefore, the study confirms the importance of these detected sites that could indicate their differences in cancer suppressing property.

Keywords: Data-mining, association network, protein sequence alignment, granular computing, bioinformatics.

1 Introduction

Biological sequences when aligned can provide the common or discriminatory information about the individual residue of the biomolecule family. It can also provide the information from which knowledge can be extracted that directs us towards the functional sites of the molecule. Identifying the relationships between the sequences and their relationship to structure and biological functionality is an active area of research (for examples, see Chiu & Kolodziejczak, 1991, Chiu & Lui, 2005, Chiu & Liu,

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 220–230, 2013.

2012). Identifying the sequence patterns that represent the functional characteristics of the biomolecule is vital in nanostructure analysis such as drug discovery (González, Liao, & Wu, 2010).

Previous studies, involving the multiple sequence alignment of related species have indicated that various kinds of interdependent or associated patterns can be indicative of the structural and functional characteristics of the biomolecule (Chiu, Chen, & Wong, 2001; Chiu & Lui, 2005; Chiu & Wong, 2004; Chiu & Lui, 2009; Chiu & Wang, 2006; Chiu & Xu, 2011). In this paper, a new method in inferring the association network in aligned sequence ensembles is proposed. It is derived from the concept of granular computing, where information is extracted at different levels of granularity or resolution (Lin et al. 1997, 2003). It involves the use of different sizes of two-dimensional contingency table analysis by focusing on the statistical associations between different outcome subsets (Chiu & Cheung 1989, Chiu et al. 1990, 1991). Furthermore, molecular sites with association patterns having multiple relationships with other sites demonstrate convergent information (Durston et al. 2012).

In the proposed analysis, there are two consecutive phases. First, the molecular sites in the multiple sequence alignment are labeled into three different types based on their site association characteristics: conserved sites (C-sites), interdependent sites (D-sites), and hypervariate or other sites (H-sites). Next, the importance of these sites is evaluated by testing their association to the functionality of the biomolecule such as known structural or functional characteristics.

In an aligned sequence ensemble, associated sites refer to sites that have statistical significance relationship with another site. In proteins, they represent sites with amino acid pairs observed together. Two types of associations can be considered, the association between two sites (such as X and Y sites) and the association among multiple sites (such as W, X, Y, and Z sites). Previous studies using multiple sequence alignment have observed that associated sites can predict the functional sites in biomolecules. For example, the patterns derived from associated sites were capable of inferring secondary and tertiary bonding structures (Chiu & Kolodziejczak, 1991), and have been used for the recognition of the ribosome binding sites in E. coli (Frishman, 1999). Similar sites can also have conformational, biochemical, and taxonomical significance (Wong, Liu, & Wang, 1996; Chiu et al., 2001). In other studies, regions obtained from statistical patterns are shown to correspond to exon sub-regions (Chiu & Lui, 2005) and the identification of the three-dimensional molecular core sites (Chiu & Lui, 2012).

2 Associations at Different Levels

One of the fundamental tasks of data mining is the discovery, description and quantification of the associations within the data (Pedrycz, 2001). Typically, the information from the associations in an event is detected considering the complete outcome space. However, the associations in the given dataset can be a global or a local phenomenon (Fig. 1). The two phenomena can be quite different and their information hence may convey different characteristics.

Figure 1 depicts a probability distribution curve with two different phenomena, local and global deviations from the expected pattern event. At the global and local levels, the observation pattern event deviates from two different null hypotheses, H^1_0 and H^2_0, respectively. At the global level, the observed data have defined deviation, whereas at the local level, the data can further deviate from the locally expected distribution.

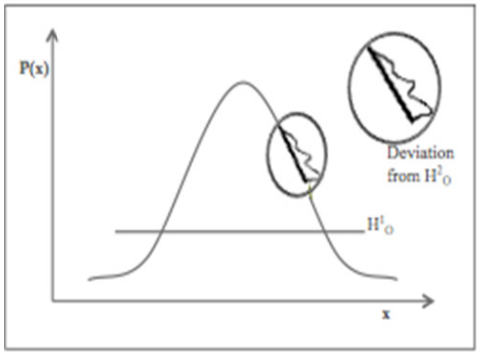


Fig. 1. Probability curve showing deviation from H^1_o and H^2_o

Hence, the information at one level of resolution may not exist at another, and this information may be significant (Chiu et al., 1991). Therefore focusing on multiple levels of resolution provides a more complete basis for data abstraction and knowledge discovery and can be extremely valuable for some datasets.

3 p53- Guardian of the Genome and Its Homologs

Lane (1992) first called the tumor suppressor protein p53 the "guardian of the genome" and Levine (1997) called it the "cellular gatekeeper". This molecule has been actively studied world-wide ever since. Under stress conditions, such as DNA damage (such as from ionizing radiation, UV radiation, and chemotherapeutic agents), heat shock, hypoxia, and oncogene over-expression, wild type p53 is activated and triggers diverse biological responses in cell cycle arrest, DNA repair, apoptosis, and cellular senescence. Hence p53 prevents the replication of damaged DNA and maintains the integrity of the genome.

The human p53 protein (Joerger & Fersht, 2007) is 393 amino acids long and has three domains: an N-terminal transactivation domain (1-93), a sequence specific DNA binding domain (102-292) and a C-terminal oligomerization domain (323-393).

The inactivation of p53 due to mutations, deletion, or interaction with cellular and viral proteins is a common event in the development of diverse types of cancer. Indeed, p53 is frequently inactivated in about 45-50% of all types of cancer (Greenblatt et al. 1994; Lane et al., 2010, Hollstein et al., 1991). Under normal conditions, the active p53 responds to the DNA damage in the cells and prevents the proliferation of damaged cells. When p53 is inactivated, it loses its biological function, permitting the proliferation of cells that carry damaged DNA, eventually leading to tumor formation.

In 1997 and 1998, p73 and p63, respectively, were identified as structural and functional homologs of p53 (Melino et al., 2003). The overall domain structure of the p53 family members is conserved and consists of a transactivation domain (TAD), DNA binding domain (DBD), and oligomerization domain (OGD). Unlike p53, the genes encoding p63 and p73 are rarely mutated in human cancer, and knock-out mice studies demonstrate developmental defects rather than a propensity for tumor formation.

4 Methodology

4.1 The First Phase of Analysis

In the first phase of our proposed analysis, the aligned sites in the p53 protein multiple sequence alignment were labeled into different types based on aligned site characteristics. The three different types of sites were also discussed in (Wong et al. 1976; Chiu & Wang, 2006):

- Associated sites (D-sites): The D-sites indicated the sites with observed amino acid values multiply associated with the values of other sites, reflecting a complex interdependent relationship.
- Invariant or conserved sites (C-sites): The C-sites indicated the sites mostly with the same amino acid value, reflecting constant value observation.
- Hypervariate sites (H-sites): The H-sites indicated the sites that could not be classified into either the D-site or the C-site types.

In multiple sequence alignment of a biomolecule, convergent association pattern (such as D-sites) represented the sites that have association relationship with other sites converging on them. The association relationship between sites was detected by using a suitable statistical hypothesis test. In an aligned ensemble, each aligned site was statistically tested for association with all other sites. In our case, when a site was found to be significantly associated with more than one site, it was considered to have a convergent association pattern that reflected a multiple interdependence relationship. For example, in Figure 2, site S3 was tested for association with all the other sites and the sites associated with S3 were indicated by the P1 (site-site) pattern.

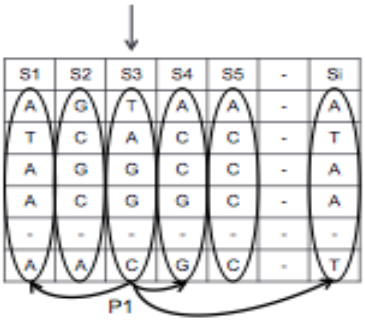


Fig. 2. Site-site pattern (P1) (modified from Chiu & Xu, 2011)

A statistical hypothesis test was used to evaluate the association relationship between two distinct sites in the aligned sequences. The goal was to evaluate whether or not a site was significantly associated with other sites in the aligned ensemble. It was hypothesized that in identifying the network of association patterns, the underlying functional structure of the biomolecules may be revealed.

4.2 Selection of Statistical Test

In general with large sample size, chi-square test can be applied to evaluate the significance of the association relationship between the site variables. Here, the sample size was small resulting in sparse contingency tables. Thus Fisher`s exact test could be applied.

4.3 Correction for Multiple Testing

In this phase, each aligned site in the alignment was tested for association relationship with all other sites. With multiple hypotheses tested, Bonferroni correction was applied to control the familywise error rate:

$$\alpha` = (\alpha)/n$$

where α is the significance level and n is the number of multiple tests.

4.4 Detection of D-sites Using Different Sizes of Contingency Tables

The use of the proposed method based on granular association facilitated the identification of D-sites in the aligned sequence ensembles for different outcome subsets between two variables. Multiple levels of data abstraction were constructed by using different sizes of the two-dimensional contingency tables. Based on three different sizes of the contingency table, three levels of analysis could be employed:

- Full contingency table analysis (R_F)
- 2x2 contingency sub-table analysis (R_{2x2})
- Single cell contingency table analysis (R_1)

4.5 Full Contingency Table Analysis (R_F Method)

The standard full contingency table analysis evaluates the association relationship between two distinct sites from an aligned sequence ensemble. After the contingency table relating two sites in aligned sequences is generated, Fisher's exact test can be applied to each relationship. The test detects the significance of the association between the two selected distinct sites. The null hypothesis is that the site variables, say X and Y, are independent and the alternate hypothesis otherwise. If the test statistic is larger than the tabulated value at a pre-defined significance level, then the association is accepted as significant.

4.6 2x2 Contingency Sub-table Analysis (R_{2x2} Method)

The 2x2 contingency sub-table analysis of a two-dimensional table evaluates the association between the outcome subsets, denoted as sub-X, and sub-Y that was selected by using relevant criteria from the full contingency table. There were two criteria for selecting a sub-table, analogous to the use of two different but similar estimators.

The first selection criterion for selecting the 2x2 sub-table can be described as follows:

- Select the first two outcomes from a full contingency table with the highest marginal frequency.
- Create a sub-table involving the human amino acid in the two sites.

The second similar selection criterion can be used:

- Select the human amino acid in the two X and Y sites.
- Select the non-human amino acid in the X and Y sites with the highest marginal frequency.

After the 2x2 sub-table is constructed, the test of independence was applied to the two sites.

4.7 Single Cell Contingency Table Analysis (R_1 Method)

With a full contingency table constructed relating between, say sites X and Y, the cell with the observed amino acid in the human sequence of site X and site Y was selected. The hypothesis test is then applied to identify significant associations. The test statistic is computed based on the normal distribution on the difference between the observed and expected frequencies (Haberman, 1973, Wong & Wang, 1997). If the test statistic is larger than the tabulated value at a pre-defined significance level, then the association is accepted as significant. In another words, the single cell contingency table analysis is applied to evaluate the association between two different sites of the human sequence based on the distribution obtained from the aligned sequence ensemble.

4.8 The Second Phase of Analysis

In the second phase, the association between the defined patterns and a targeted functional characteristic of the p53 protein is evaluated.

As described before, the different types of statistical patterns can be classified into seven different categories:

- Conserved sites pattern (CS): It indicates sites with mostly a constant value observation.
- R_{2x2} pattern: It indicated sites identified as significantly associated using the 2x2 contingency sub-table method.
- R_1 pattern: It indicated sites identified as significantly associated sites by the single cell contingency table method.

- CS + R_{2x2} pattern: It indicated sites that are either conserved or identified as associated sites by the 2x2 contingency sub-table method.
- CS + R_1 pattern: It indicated sites that are either conserved or identified as significantly associated sites by the single cell contingency table method.
- R_{2x2} + R_1 pattern: It indicated sites identified as significantly associated sites by either the 2x2 contingency sub-table or the single cell contingency table method.
- CS + R_{2x2} + R_1 pattern: It indicated sites that are either conserved or identified as significantly associated sites by the 2x2 contingency sub-table or the single cell contingency table method.

The goal here is to analyze the association between the identified patterns and targeted functionalities to determine if they are significantly associated. This analysis would be useful in identifying significant functional association, possibly leading to the discovery of specific functional sites with the desirable properties. In the experiments, we had considered six different p53 functionalities, including structural characteristics and amino acid differences between p53 and its homologs of p63 and p73. There are five different types of discrimination between p53, p63 and p73, as:

- Type I: The amino acid in the human sequence of p53, p63, and p73 are observed the same.
- Type II: The amino acid in the human sequence of p53, p63, and p73 are observed different.
- Type III: The amino acid in the human sequence of p53 observed differently from that of p63 and p73.
- Type IV: The amino acid in the human sequence of p63 observed differently from that of p53 and p73.
- Type V: The amino acid in the human sequence of p73 observed differently from that of p53 and p63.

4.9 Test of Independence in the Second Phase of Analysis

The statistical significance between the generated site patterns and the functional characteristics is evaluated using a test of independence from the construction of a new 2x2 contingency table, indicating whether the pattern and the functionality are significantly associated or not. The variable on the rows in the table indicated a targeted functionality (e.g. polarity) and the variable on the columns indicated the generated site pattern (e.g. CS pattern). The chi-square statistical test is then applied.

The null hypothesis assumes that the pattern (P) and the functionality (F) are independent and the alternate hypothesis otherwise. From the observed frequency table, the observed and expected frequencies are then calculated. The chi-square statistic is computed with one degree of freedom based on the deviations between the observed frequencies from the expected frequencies. The association relationship between the variables P and F is considered to be statistically significant if $\chi 2 > N_\alpha$, where N_α was the tabulated threshold value with one degree of freedom and α is the confidence level.

5 Experimental Studies Using the p53 Protein Alignment

The amino acid sequences used in the experiments were obtained from the UniProtKB database (http://www.uniprot.org). The database stored 34 different species of p53 sequences, three species of p63 sequences and 3 sequences of p73 sequences.

In the first phase of analysis, the multiple sequence alignment of 34 p53 sequences was obtained, using the alignment from the ClustalW (Version 2.1) program. The following ClustalW default settings were used. (The pairwise alignment parameters were: protein weight matrix = Gonnet, gap open penalty = 10, and gap extension penalty = 0.1; the multiple alignment parameters were: protein weight matrix = Gonnet, gap open penalty = 10, gap extension penalty = 0.2, gap separation distances = 5, end gaps = off, and clustering method = neighbor joining.) The alignment indicated 115 sites as conserved sites and these sites were labeled as C-sites. The remaining 278 (393-115) aligned sites were employed in the experiments, to identify the D-sites and the H-sites.

The three levels of data abstraction methods, R_F, R_{2x2}, and R_1, were applied generating the labeled sites (as D-sites). Due to the small sample size of the data and R_F generates largely sparse contingency tables, hence the method were excluded from further analysis.

In the R_{2x2} method, two selection criteria (as two estimators) were used to select a sub-table from a full contingency table. In the p53-aligned data, it was found that both criteria selected similar D-sites as expected.

The proposed R_{2x2} method identified 107 D-sites with a 5% significance level after using the Bonferroni correction. In the transactivation domain (TAD), DNA binding domain (DBD), and oligomerization domain (OD), there were 20, 52, and 34 sites identified respectively.

The R_1 method identified 28 D-sites with a 5% significance level after using the Bonferroni correction. In the transactivation domain (TAD), DNA binding domain (DBD), and oligomerization domain (OD), there were 20, 4, and 4 sites identified respectively.

In the second phase of analysis, the human sequences of p53, p63, and p73 were aligned according to their common domains. This alignment was used to identify the discriminating types between the p53 family members.

The five different types used to discriminate among the human sequences of p53, p63, and p73 molecules were described. The association relationship between the defined patterns and the discriminating types were analyzed. The number of D-sites selected in type III was high in both the R_{2x2} and R_1 methods. Since type III differentiated p53 from the other two family members of p63 and p73. This relationship between the defined patterns was the most important.

The observed chi-square values and p-values for association testing between each pattern and discriminate type III were noted. Figure 3 shows that D-sites are mostly associated with type III (which discriminate between p53 and its homologs). The frequencies clearly demonstrated that the patterns CS, R_{2x2}, CS + R_1, and R_{2x2} + R_1 were stronger and statistically significant with type III discrimination with 0.01% significance level. The R_{2x2} + R_1 pattern was more significant than the individual effect of either R_{2x2} or R_1. However, when the CS pattern was considered with the other patterns (CS + R_{2x2}, and CS + R_{2x2} + R_1), the chi-square value decreased drastically and was also weaker. The results can be interpreted as follows:

- The patterns, CS, R_{2x2}, CS + R_1, and R_{2x2} + R_1, had different effect in discriminating between p53 and p63/p73.
- When the patterns, CS, R_{2x2}, and R_1, were considered together, the effects cancelled each other or that the CS pattern had an interactive effect with the D-sites effect.

6 Discussions and Conclusions

The experimental studies on p53 protein multiple sequence alignment confirm that the proposed granular association evaluation method is useful to identify and label associated site patterns. The method extracts information based on an outcome subspace in the data by using different resolutions (or sizes) of the two-dimensional contingency table. The experiments on p53 showed that the method identifies associated patterns in the R_{2x2} and R_1 analyses. Also, the R_{2x2} method identifies a higher number of sites than the R1 method, and the number of sites associated with each site may differ. The second phase of analysis revealed that the defined patterns can be associated with some targeted structural and functional properties of the p53 protein. In summary, the extracted association patterns have proven to be useful in discovering sites with some structural and functional properties of a protein molecule.

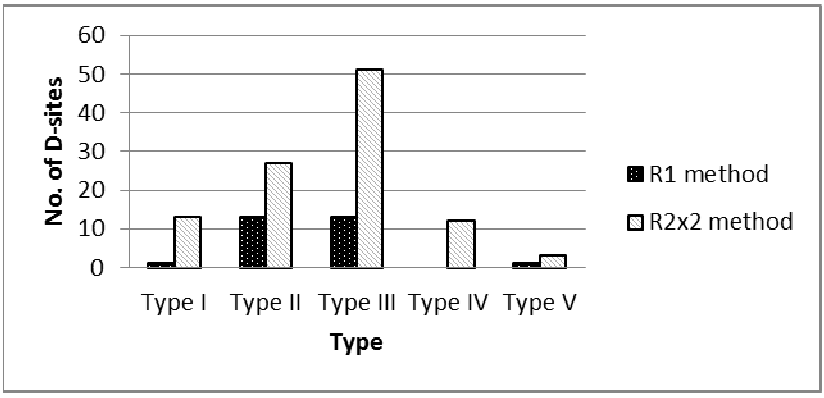


Fig. 3. Number of D-sites in discriminating among the human sequence of p53, p63 and p73. Note D-sites most distinguish p53 from its homologs (as in Type III).

Acknowledgements. The research is supported by Natural Sciences and Engineering Research Council of Canada, Discovery Grant #400297.

References

1. Chiu, D.K.Y., Chen, X., Wong, A.K.C.: Association Between Statistical and Functional Patterns in Biomolecules. In: Proceedings of the Atlantic Symposium on Computational Biology and Genome Information Systems and Technology, Durham, USA, pp. 64–69 (2001)

2. Chiu, D.K.Y., Cheung, B.: Hierarchical Maximum Entropy Discretization. Computing and Information. In: Proceedings of the International Conference on Computing and Information (ICCI 1989), pp. 237–242. North-Holland, Toronto (1989)
3. Chiu, D.K.Y., Cheung, B., Wong, A.K.C.: Information Synthesis based on Hierarchical Maximum Entropy Discretization. Journal of Experimental and Theoretical Artificial Intelligence 2, 117–129 (1990)
4. Chiu, D.K.Y., Kolodziejczak, T.: Inferring Consensus Structure from Nucleic Acid Sequences. Computational Applications in Biosciences 7, 347–352 (1991)
5. Chiu, D.K.Y., Lui, T.W.H.: NHOP: A Nested Associative Pattern for Analysis of Consensus Sequence Ensembles. IEEE Trans. on Knowledge and Data Engineering (2012) (in press)
6. Chiu, D.K.Y., Lui, T.W.H.: A Multiple-pattern Biosequence Analysis Method for Diverse Source Association Mining. Applied Bioinformatics 4(2), 85–92 (2005)
7. Chiu, D.K.Y., Wang, Y.: Multipattern Consensus Regions in Multiple Aligned Protein Sequences and their Segmentation. EURASIP J. Bioinformatics Syst. Biol. 35809, 1–8 (2006)
8. Chiu, D.K.Y., Wong, A.K.C.: Multiple Pattern Associations for Interpreting Structural and Functional Characteristics of Biomolecules. Information Science 167, 23–39 (2004)
9. Chiu, D.K.Y., Wong, A.K.C., Cheung, B.: Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 126–140. MIT Press, Cambridge (1991)
10. Chiu, D.K.Y., Xu, P.S.C.: InfoBarcoding: Selection of Non-Contiguous Sites in Molecular Biomarker. In: Proceeding, Computational Advances in Bio. and Medical Sciences (ICCABS), pp. 69–74 (2011)
11. Durston, K., Chiu, D.K.Y., Wong, A.K.C., Li, G.C.L.: Statistical Discovery of Site Inter-Dependencies in Sub-Molecular Hierarchical Protein Structuring. EURASIP J. on Bioinformatics and Systems Biology 2012, 8 (2012)
12. European Bioinformatics Institute tool for Multiple Sequence Alignment using clustalw2, `http://www.ebi.ac.uk/Tools/msa/clustalw2.html/`
13. Frishman, D., Mironov, A., Gelfand, M.: Starts of Bacterial Genes: Estimating the Reliability of Computer Predictions. Gene 234, 257–265 (1999)
14. Gonzalez, A.J., Liao, L., Wu, C.H.: Predicting Ligand-Binding Residues using Multi-Positional Correlations and Kernel Canonical Correlation Analysis. In: Proc. 2010 IEEE Intern. Conf. of Bioinformatics and Biomedicine (BIBM), pp. 158–163 (2010)
15. Greenblatt, M.S., Bennett, W.P., Hollstein, M., Harris, C.C.: Mutations in the p53 Tumor Suppressor Gene: Clues to Cancer Etiology and Molecular Pathogenesis. Cancer Research 54, 4855–4878 (1994)
16. Haberman, S.J.: The Analysis of Residuals in Cross-Classified Tables. Biometrics 29, 205–220 (1973)
17. Hollstein, M., Sidransky, D., Vogelstein, B., Harris, C.C.: p53 Mutations in Human Cancers. Science 253(5015), 49–53 (1991)
18. Joerger, A.C., Fersht, A.R.: Structural Biology of the Tumor Suppressor p53 and Cancer-Associated Mutants. Advanced Cancer Research 97, 1–23 (2007)
19. Lane, D.P.: Cancer and p53, Guardian of the Genome. Nature 358, 15–16 (1992)
20. Lane, D.P., Cheok, C.F., Lain, S.: p53-based Cancer Therapy. Cold Spring Harb. Perspect. Biol., 2, a001222 (2010)
21. Lin, T.Y.: Granular computing. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 16–24. Springer, Heidelberg (2003)

22. Lin, T.Y.: From Rough Sets and Neighborhood Systems to Information Granulation and Computing in Words. In: European Congress on Intelligent Techniques and Soft Computing, pp. 1602–1607 (1997)
23. Melino, G., Lu, X., Gasco, M., Crook, T., Knight, R.A.: Functional Regulation of p73 and p63: Development and Cancer. Trends Biochem. Sci. 28, 663–670 (2003)
24. Pedrycz, W.: Granular Computing: An Emerging Paradigm. Physica-Verlag, Heidelberg (2003)
25. The p53 website, http://p53.free.fr/
26. The UniProtKB database, http://www.uniprot.org
27. Wong, A.K.C., Lui, T.S., Wang, C.C.: Statistical Analysis of Residue Variability in Cytochrome C. J. Molecular Biology 102(2), 287–295 (1976)
28. Wong, A.K.C., Wang, Y.: High-Order Pattern Discovery from Discrete-Valued Data. IEEE Trans. on Knowledge Systems 9(6), 877–893 (1997)

Prediction of Non-genotoxic Hepatocarcinogenicity Using Chemical-Protein Interactions

Chun-Wei Tung

[1] School of Pharmacy, Kaohsiung Medical University, 807, Taiwan
[2] Ph.D. Program in Toxicology, Kaohsiung Medical University, 807, Taiwan
cwtung@kmu.edu.tw
http://cwtung.kmu.edu.tw

Abstract. The assessment of non-genotoxic hepatocarcinogenicity of chemicals is currently based on 2-year rodent bioassays. It is desirable to develop a fast and effective method to accelerate the identification of potential hepatocarcinogenicity of non-genotoxic chemicals. In this study, a novel method CPI is proposed to predict potential hepatocarcinogenicity of non-genotoxic chemicals. The CPI method is based on chemical-protein interactions and interpretable decision tree classifiers. The interpretable rules generated by the CPI method are analyzed to provide insights into the mechanism and biomarkers of non-genotoxic hepatocarcinogenicity. The CPI method with an independent test accuracy of 86% using only 1 protein biomarker outperforms the state-of-the-art methods of gene expression profile-based toxicogenomics using 90 gene biomarkers. A protein ABCC3 was identified as a potential protein biomarker for further exploration. This study presents the potential application of CPI method for assessing non-genotoxic hepatocarcinogenicity of chemicals.

Keywords: Non-Genotoxic Hepatocarcinogenicity, Decision Tree, Chemical-Protein Interaction, Interpretable Rule, Toxicology.

1 Introduction

Chemical carcinogenesis can be classified into two main categories of genotoxic (mutagenic) and non-genotoxic (non-mutagenic) agents according to the mechanism of action [1, 2]. Several short-term *in vitro* and *in vivo* assays have been developed to assess genotoxic agents by measuring DNA damage, mutagenic effects, and chromosomal aberrations [3]. However, due to the complex nature of non-genotoxic agents, the assessment of non-genotoxic hepatocarcinogenicity of chemical compounds is based on 2-year rodent bioassays that is labor-intensive, time-consuming and expensive. There are only 1500 chemicals studied by National Toxicology Program during the past 30 years [4]. It is desirable to develop alternative methods to efficiently prioritize potential non-genotoxic hepatocarcinogenicity of chemicals for further studies.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 231–241, 2013.

Numerous computational models have been developed to predict various toxicity endpoints with reasonably good prediction performance. For example, the quantitative structure-activity relationship (QSAR) models have been extensively used to analyze and predict carcinogenicity [5–8]. QSAR model aiming to correlate chemical structure information and toxicity endpoints could provide useful information of important structure for toxicity alerts. However, the application of QSAR models for predicting non-genotoxic hepatocarcinogenicity yields a low accuracy of 55% [9] showing the complexity of non-genotoxic hepatocarcinogenicity.

Recently, toxicogenomics (TGx) correlating gene expression profiles and toxicity endpoints has emerged as important alternative methods. With the power of machine learning methods, TGx performs well in non-genotoxic hepatocarcinogenicity with a test accuracy of 80% [9, 10]. In contrast to traditional 2-year rodent bioassays, TGx methods require much less experimental effort. Generally, published TGx methods select 29 to 120 genes as important biomarkers and require short-term experiments with 5 to 28 days [9, 11, 10]. However, compared to the pure computational method QSAR, TGx methods are still more time-consuming and expensive. Also, chemical-protein interaction (CPI) as an important mechanism for toxicity can not be modeled by TGx methods. The development of fast and accurate methods can largely help the assessment of non-genotoxic hepatocarcinogenicity of chemicals.

The data of CPI information grows very fast in recent years. Benefit from the development of CPI databases, enormous interaction data obtained from databases, experiments and text-mining can be easily accessed from the structured databases including STITCH [12–14], ChemProt [15, 16] and CTD [17]. The databases makes it possible to develop a CPI-based method for analyzing and predicting non-genotoxic hepatocarcinogenicity.

In this study, a CPI based classification method is proposed to analyze and predict non-genotoxic hepatocarcinogenicity of chemicals. Decision tree algorithms capable of generating rule-based knowledge are applied to construct prediction classifiers. The 5-fold cross-validation and independent test accuracies on training and independent test dataset using only one protein are 82% and 86%, respectively. The independent test accuracy of the proposed CPI method is better than that of TGx methods requiring 1 to 5-day experiments and 90 biomarkers. This is the first study that utilizes chemical-protein interaction data to predict non-genotoxic hepatocarcinogenicity of chemicals.

2 Materials and Methods

2.1 Dataset

In this study, the development of datasets is based on a liver cancer database NCTRlcdb [18]. In order to demonstrate and compare prediction performances of different methods including CPI, QSAR and toxicogenomics models, only chemicals with existing gene expression data in rat were selected from NCTRlcdb.

A final dataset consisting of 62 chemicals is utilized to develop and test classifiers for non-genotoxic hepatocarcinogenicity that is developed by Liu *et al.* [9]. Class labels of either liver carcinogens, carcinogens in other organisms, or noncarcinogens for chemicals were obtained from NCTRlcdb. In order to compare with the QSAR and toxicogenomics models of the previous study [9], the 62 chemicals are divided into a training dataset and an independent test dataset according to the previous study [9]. The training and independent test datasets consisting of 8 positive and 32 negative chemicals and 5 positive and 17 negative chemicals are utilized for training and testing models, respectively.

2.2 Chemical-Protein Interactions

Chemical-protein interaction data are obtained from STITCH 3.1 database [13, 14, 12]. STITCH database is an aggregated database of interactions connecting over 300,000 chemicals and 2.6 million proteins from 1133 organisms. The interaction data are obtained from three major sources of experiments, databases and text-mining. The experiment part consists of direct chemical-protein binding data with experimental evidence. The database part contains interaction data from pathway databases. The text-mining data is obtained by extracting information of interactions from literatures using text-mining techniques. Likelihood or relevance scores of interactions are available for each evidence type. An overall score for a given chemical-protein interaction is generated by combining three scores of corresponding evidence types that is available at STITCH [19]. The score is a integer value ranging from 0 (no interaction) to 1000 (strong interaction). Chemical-protein interactions are transferred between species based on the sequence similarity of the proteins [19].

2.3 Decision Tree Algorithm

Decision tree algorithms capable of generating interpretable rules based on training data are widely used in various classification and regression problems such as immunogenic peptides [20], ubiquitination sites [21], gamma-turn types [22] and protein subnuclear localization [23]. In this study, the decision tree method C5.0 is applied to construct decision tree classifiers and derive interpretable rules based on chemical-protein interaction profile for classifying non-genotoxic hepatocarcinogenicity. C5.0 is an improved version of C4.5 with smaller trees and less computation time [24]. The implementation of C5.0 used in this study is the R package C50 [25].

The construction of a decision tree is described as follows. First, information gain is utilized to rank features. Second, the top-ranking features are iteratively appended as nodes to split data into subsets. The tree growing process stops when the data subset in each leaf node belongs to the same class. The fully-grown tree is prone to over-fit the training data. Therefore, a pruning process is applied to reduce the tree size by replacing a subtree with a leaf node to avoid over-fitting problems. The pruning process is based on a default threshold

value of 25% confidence. The samples in the leaf node are the covered samples of this rule. The class label of a leaf node is determined by using majority rule. The samples with a relative small size in the leaf node are regarded as misclassified samples. The final decision tree can directly generate if-then rules where one leaf node corresponds to one rule.

2.4 Performance Measurement

To evaluate classifiers for their prediction performance, the widely used 5-fold cross-validation method is applied. Four measurements were applied to evaluate classifiers including sensitivity, specificity, accuracy and Matthews correlation coefficient (MCC) defined as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{1}$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \tag{2}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}, \tag{3}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{4}$$

where *TP*, *FP*, *FN* and *TN* are the numbers of true positives, false positives, false negatives and true negatives, respectively. In this work, accuracy is used as major indicator for estimating the performance of classifiers.

3 Results and Discussion

3.1 Classification Performance on Training Dataset

The proposed CPI method is based on information of chemical-protein interactions. The chemical of *N,N'*-diphenyl-*p*-phenylenediamine without a corresponding record in STITCH database is excluded from the following analyses. The chemicals in the training dataset is firstly transformed to 4 matrixes of chemical-protein interaction scores obtained from combined scores, databases, experiments and text-mining. Only the CPI information of *Rattus norvegicus* is used because the hepatocarcinogenic annotation of the 62 chemicals is based on rat and mouse. In order to provide better insights into protein biomarkers of non-genotoxic hepatocarcinogenicity, the decision tree algorithm C5.0 is applied to generate human interpretable rules based on training datasets for further confirmation.

Table 1. Cross-validation performance

Model type	Classifier	Feature selection	Number of selected features	5-CV accuracy
CPI	C5.0	Information gain	1	0.82
QSAR*	NCC	Wrapper-based mRMR	15	0.76
TGx (1-day)*	NCC	Wrapper-based mRMR	90	0.87
TGx (3-day)*	NCC	Wrapper-based mRMR	90	0.87
TGx (5-day)*	NCC	Wrapper-based mRMR	90	0.90

* Model performance from Liu *et al* [9]

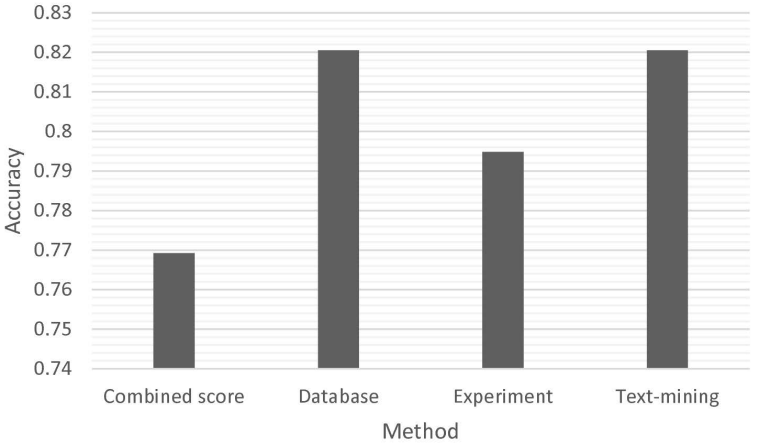


Fig. 1. Five-fold cross-validation performance of CPI method using various scores as features

To evaluate the classification performance of the CPI method, a 5-fold cross-validation (5-CV) is applied to the training dataset consisting of 7 positive and 31 negative chemicals. In the 5-CV, the training dataset is firstly divided into 5 folds with nearly equal number of chemicals. For each validation fold f of the 5-CV, C5.0 is applied to select important features for constructing a decision tree classifier based on the remaining 4 folds and evaluate its performance on the validation fold. The 5-CV performances of the CPI method for 4 matrixes are shown in Fig. 1. The CPI scores obtained from databases and text-mining perform best with the same accuracy of 82.05%. The accuracy of experiment-derived CPI scores is slightly worse with an accuracy of 79.49%. The CPI scores obtained by combining three data sources of databases, experiments and text-mining perform worst with an accuracy of 76.92%.

The information obtained from databases including metabolic pathway information is used for the following analysis that could be more useful than information from text-mining because chemical metabolites might be more toxic

than parent chemicals. Table 1 shows the detailed 5-CV performance of the CPI method and published QSAR and TGx methods [9]. All the QSAR and TGx methods are based on a nearest-centroid classifier (NCC) and a wrapper-based feature selection based on the ranking calculated by a minimum redundancymaximum relevancy (mRMR) [26]. The CPI method utilizing a simple and human interpretable classifier C5.0 shows good accuracy of 0.82 that is better than the QSAR model. Although TGx models show better accuracies than CPI, its feature selection method is a wrapper-based method that is more likely to overfit the training dataset and overestimate its prediction performance. Additionally, the proposed CPI method utilizes only 1 feature for each fold with interpretable rules that is much smaller than the QSAR and TGx models requiring 15 and 90 features without interpretable rules, respectively. The selected features will be discussed in the next section.

3.2 Feature Selection of Important Proteins

For each fold of the 5-fold cross-validation, C5.0 select important features for constructing a decision tree classifier. The interpretation of the decision tree classifier can provide better understanding of non-genotoxic hepatocarcinogenicity. The important features of the five decision trees are shown in Table 2 with a usage value showing the percentage of covered chemicals.

Due to the simple decision tree created for each fold with only one protein, all the usage values are 100%. The ABCC3 protein is identified as an important protein in two folds (40%) showing its critical role in non-genotoxic hepatocarcinogenicity. ABCC3 (ATP-binding cassette, subfamily C (CFTR/MRP), member 3) is a member of the superfamily of ATP-binding cassette (ABC) transporters that transports various molecules across membranes. ABCC3, also known as the canalicular multispecific organic anion transporter 2, exhibits drug transmembrane transporter activity that is critical for drug transport, multidrug resistance and bile acid transport pathways. The rule associated with ABCC3 is 'IF a chemical interacts with ABCC3 THEN it is a hepatocarcinogenic chemical'.

The protein MPO is a myeloperoxidase with peroxidase activity and is found in extracellular space, mitochondrion and secretory granule. Previous studies have reported possible roles of oxidative stress on carcinogenicity [27, 28]. MPO as an antioxidant enzyme is able to detoxify the reactive oxygen species (ROS) of oxidative stress. Chemicals interacting with MPO could interrupt the detoxification process and lead to carcinogenicity.

Serotransferrin (TF) exhibiting the activity of binding and transmembrane transporter of ferric iron is identified in the third fold. Iron in its free form is carcinogenic unless it is bound to ferritin or transferrin [29–31]. The carcinogenicity of TF-interacting chemicals might be caused by their interference with the loading of iron.

The protein RB1 of retinoblastoma 1 associated with retinoblastoma is found to be involved in the non-genotoxic hepatocarcinogenicity [32]. RB1 is a tumor suppressor protein for preventing excessive cell growth by inhibiting cell cycle progression [33]. The dysfunction of RB1 could cause carcinogenicity.

Table 2. Important proteins identified from 5-fold cross-validation

Fold	Name	Description	Usage
1	ABCC3	ATP-binding cassette, subfamily C (CFTR/MRP), member 3	100%
2	MPO	Myeloperoxidase	100%
3	TF	Serotransferrin	100%
4	RB1	Retinoblastoma 1	100%
5	ABCC3	ATP-binding cassette, subfamily C (CFTR/MRP), member 3	100%

Altogether, the identified proteins and functions are consistent with possible mechanisms of non-genotoxic carcinogenicity reported by previous studies, including modulation of metabolic enzymes, induction of peroxisome proliferation and alteration of intercellular communication [34–37].

3.3 Independent Test

To further evaluate the prediction ability of the CPI method, the proposed CPI method is applied to train a decision tree classifier based on the training dataset and predict the independent test dataset consisting of 20 chemicals. A search of the chemical of lead(iv) acetate in STITCH database leads to the record of lead(ii) acetate of the same CPI profiles. To avoid overestimate the prediction performance of the CPI method, the chemical of lead(iv) acetate is excluded for the following analysis. The same as the 5-CV with 1 protein selected for each fold, only 1 protein is selected to construct a decision tree classifier. The decision tree shown in Fig. 2 represents a very simple rule of 'IF a chemical interacts with ABCC3 THEN it is a hepatocarcinogenic chemical'. The rule is surprisingly simple and correctly predict 90% chemicals in the training dataset with only 4 misclassified chemicals. All 31 non-hepatocarcinogenic chemicals do not interact with protein ABCC3. Fifty percent of hepatocarcinogenic chemicals interact with ABCC3. Chemicals interact with ABCC3 might interfere the normal function of chemical transportation.

Table 3. Independent test performance

Model type	Number of selected features	Accuracy	Sensitivity	Specificity	MCC
CPI	1	0.86	0.40	1.00	0.580
QSAR*	15	0.55	0.20	0.65	-0.138
TGx (1-day)*	90	0.77	0.40	0.88	0.307
TGx (3-day)*	90	0.77	0.20	0.94	0.206
TGx (5-day)*	90	0.82	0.60	0.88	0.482

* Model performance from Liu *et al* [9]

To demonstrate the prediction ability of the proposed CPI method, the decision tree classifier is applied to predict chemicals in the independent test dataset. The prediction results are shown in Table 3. The simple decision tree classifier

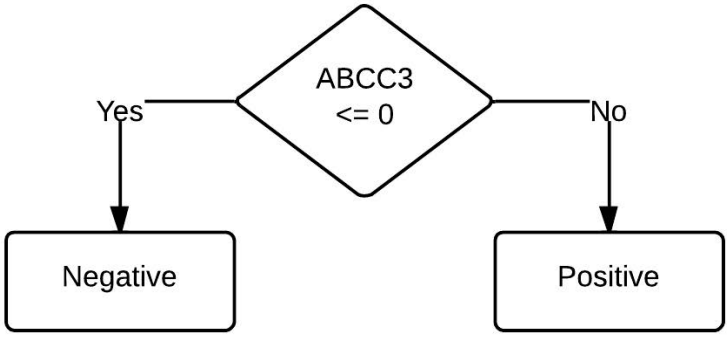


Fig. 2. The constructed decision tree based on the training dataset

of CPI performs very well with an accuracy of 86% that is better than QSAR, 1-day TGx, 3-day TGx and 5-day TGx models with accuracies of 55%, 77%, 77% and 82%. The MCC value as a more objective evaluation of performance for unbalanced data is also used to evaluate prediction performance. The MCC values for CPI, QSAR, 1-day TGx, 3-day TGx and 5-day TGx models are 0.580, -0.138, 0.307, 0.206 and 0.482, respectively. The CPI method with highest MCC value performs best.

The wrapper-based feature selection method used in the previous study [9] might overestimate the 5-CV accuracies on the training dataset and result in a large decrease in prediction accuracies on the independent test dataset. The proposed CPI method utilizing only a single feature with human interpretable rules outperforms QSAR and TGx methods showing that chemical-protein interactions are useful for predicting non-genotoxic hepatocarcinogenicity of chemicals.

4 Conclusions

Alternative methods for assessing non-genotoxic hepatocarcinogenicity of chemicals could save a lot of time and money and reduce the consumption of animals for testing. The traditional QSAR model is not effective in discrimination of hepatocarcinogenicity of non-genotoxic chemicals [9] showing the complex nature of non-genotoxic hepatocarcinogenicity involving many genes and proteins. In contrast to chemical structure-based QSAR models, TGx methods based on gene expression-profiles can model the complex mechanism in the transcriptomics level and perform better than the QSAR model [9].

The mechanism of action of non-genotoxic hepatocarcinogenicity might involve complex regulations of proteins and chemicals. Hence, the application of CPI data for developing classifiers is expected to outperform QSAR and TGx methods. This study presents a novel CPI-based method and demonstrates the effectiveness of biomarker identification and superior prediction performance. The utilization of simple decision tree algorithms generates human-interpretable rules for better understanding of key proteins for non-genotoxic hepatocarcinogenicity.

The identified proteins could serve as important biomarkers for further applications to the assessment of non-genotoxic hepatocarcinogenicity of chemicals. Compared to TGx methods requiring assessment of 100 gene expression values and 5 to 28-day experiments, the identified single biomarker could be more cost-effective and time-saving. Future works include the application of advanced machine learning algorithms such as support vector machines and collection of a larger dataset for improving prediction accuracy.

Acknowledgement. The authors would like to acknowledge the financial support from National Science Council of Taiwan (NSC 101-2311-B-037-001-MY2) and Kaohsiung Medical University Research Foundation (KMU-Q110015 and KMU-ER-013).

References

1. Hayashi, Y.: Overview of genotoxic carcinogens and non-genotoxic carcinogens. Exp. Toxicol. Pathol. 44, 465–471 (1992)
2. Melnick, R.L., Kohn, M.C., Portier, C.J.: Implications for risk assessment of suggested nongenotoxic mechanisms of chemical carcinogenesis. Environ. Health Perspect. 104 (suppl. 1), 123–134 (1996)
3. Weisburger, J.H., Williams, G.M.: The distinction between genotoxic and epigenetic carcinogens and implication for cancer risk. Toxicol. Sci. 57, 4–5 (2000)
4. Gold, L.S., Manley, N.B., Slone, T.H., Rohrbach, L., Garfinkel, G.B.: Supplement to the carcinogenic potency database (cpdb): results of animal bioassays published in the general literature through 1997 and by the national toxicology program in 1997-1998. Toxicol. Sci. 85, 747–808 (2005)
5. Kar, S., Deeb, O., Roy, K.: Development of classification and regression based qsar models to predict rodent carcinogenic potency using oral slope factor. Ecotoxicol. Environ. Saf. 82, 85–95 (2012)
6. Kar, S., Roy, K.: First report on development of quantitative interspecies structure-carcinogenicity relationship models and exploring discriminatory features for rodent carcinogenicity of diverse organic chemicals using oecd guidelines. Chemosphere 87, 339–355 (2012)
7. Tanabe, K., Kurita, T., Nishida, K., Lucic, B., Amic, D., Suzuki, T.: Improvement of carcinogenicity prediction performances based on sensitivity analysis in variable selection of svm models. SAR QSAR Environ. Res. (2013)
8. Yuan, J., Pu, Y., Yin, L.: Qsar study of liver specificity of carcinogenicity of n-nitroso compounds. Ecotoxicol. Environ. Saf. 84, 282–292 (2012)
9. Liu, Z., Kelly, R., Fang, H., Ding, D., Tong, W.: Comparative analysis of predictive models for nongenotoxic hepatocarcinogenicity using both toxicogenomics and quantitative structure-activity relationships. Chem. Res. Toxicol. 24, 1062–1070 (2011)
10. Yamada, F., Sumida, K., Uehara, T., Morikawa, Y., Yamada, H., Urushidani, T., Ohno, Y.: Toxicogenomics discrimination of potential hepatocarcinogenicity of non-genotoxic compounds in rat liver. J. Appl. Toxicol. (2012)
11. Uehara, T., Hirode, M., Ono, A., Kiyosawa, N., Omura, K., Shimizu, T., Mizukawa, Y., Miyagishima, T., Nagao, T., Urushidani, T.: A toxicogenomics approach for early assessment of potential non-genotoxic hepatocarcinogenicity of chemicals in rats. Toxicology 250, 15–26 (2008)

12. Kuhn, M., von Mering, C., Campillos, M., Jensen, L.J., Bork, P.: Stitch: interaction networks of chemicals and proteins. Nucleic Acids Res. 36, D684–D688 (2008)
13. Kuhn, M., Szklarczyk, D., Franceschini, A., Campillos, M., von Mering, C., Jensen, L.J., Beyer, A., Bork, P.: Stitch 2: an interaction network database for small molecules and proteins. Nucleic Acids Res. 38, D552–D556 (2010)
14. Kuhn, M., Szklarczyk, D., Franceschini, A., von Mering, C., Jensen, L.J., Bork, P.: Stitch 3: zooming in on protein-chemical interactions. Nucleic Acids Res. 40, D876–D880 (2012)
15. Kim Kjaerulff, S., Wich, L., Kringelum, J., Jacobsen, U.P., Kouskoumvekaki, I., Audouze, K., Lund, O., Brunak, S., Oprea, T.I., Taboureau, O.: Chemprot-2.0: visual navigation in a disease chemical biology database. Nucleic Acids Res. 41, 464–469 (2013)
16. Taboureau, O., Nielsen, S.K., Audouze, K., Weinhold, N., Edsgard, D., Roque, F.S., Kouskoumvekaki, I., Bora, A., Curpan, R., Jensen, T.S., Brunak, S., Oprea, T.I.: Chemprot: a disease chemical biology database. Nucleic Acids Res. 39, D367–D372 (2011)
17. Mattingly, C.J., Colby, G.T., Forrest, J.N., Boyer, J.L.: The comparative toxicogenomics database (ctd). Environ. Health Perspec.t 111, 793–795 (2003)
18. Young, J., Tong, W., Fang, H., Xie, Q., Pearce, B., Hashemi, R., Beger, R., Cheeseman, M., Chen, J., Chang, Y.C., Kodell, R.: Building an organ-specific carcinogenic database for sar analyses. J. Toxicol. Environ. Health A 67, 1363–1389 (2004)
19. von Mering, C., Jensen, L.J., Snel, B., Hooper, S.D., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M.A., Bork, P.: String: known and predicted protein-protein associations, integrated and transferred across organisms. Nucleic Acids Res. 33, D433–D437 (2005)
20. Tung, C.W., Ho, S.Y.: Popi: predicting immunogenicity of mhc class i binding peptides by mining informative physicochemical properties. Bioinformatics 23, 942–949 (2007)
21. Tung, C.W., Ho, S.Y.: Computational identification of ubiquitylation sites from protein sequences. BMC Bioinformatics 9, 310 (2008)
22. Liaw, C., Tung, C.W., Ho, S.J., Ho, S.Y.: Sequence-based prediction of gamma-turn types using a physicochemical property-based decision tree method. Proceeding of World Academy of Science, Engineering and Technology 41, 898–902 (2010)
23. Huang, W.L., Tung, C.W., Ho, S.W., Ho, S.Y.: Proloc-rgo: Using rule-based knowledge with gene ontology terms for prediction of protein subnuclear localization. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2008, pp. 201–206. IEEE (2008)
24. Quinlan, J.: C4. 5: programs for machine learning. Morgan kaufmann (1993)
25. Kuhn, M., Weston, S.: code for C5.0 by R. Quinlan, N.C.C.: C50: C5.0 Decision Trees and Rule-Based Models (2012); R package version 0.1.0-013
26. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. 3, 185–205 (2005)
27. Kryston, T.B., Georgiev, A.B., Pissis, P., Georgakilas, A.G.: Role of oxidative stress and dna damage in human carcinogenesis. Mutat. Res. 711, 193–201 (2011)
28. Ziech, D., Franco, R., Georgakilas, A.G., Georgakila, S., Malamou-Mitsi, V., Schoneveld, O., Pappa, A., Panayiotidis, M.I.: The role of reactive oxygen species and oxidative stress in environmental carcinogenesis and biomarker development. Chem. Biol. Interact. 188, 334–339 (2010)
29. Ponka, P., Beaumont, C., Richardson, D.R.: Function and regulation of transferrin and ferritin. Semin. Hematol. 35, 35–54 (1998)

30. McCord, J.M.: Iron, free radicals, and oxidative injury. Semin. Hematol. 35, 5–12 (1998)
31. Linn, S.: Dna damage by iron and hydrogen peroxide in vitro and in vivo. Drug Metab. Rev. 30, 313–326 (1998)
32. Gill, J.H., Brickell, P., Dive, C., Roberts, R.A.: The rodent non-genotoxic hepatocarcinogen nafenopin suppresses apoptosis preferentially in non-cycling hepatocytes but also elevates cdk4, a cell cycle progression factor. Carcinogenesis 19, 1743–1747 (1998)
33. Weinberg, R.A.: The retinoblastoma protein and cell cycle control. Cell 81, 323–330 (1995)
34. Butterworth, B.E., Bogdanffy, M.S.: A comprehensive approach for integration of toxicity and cancer risk assessments. Regul. Toxicol. Pharmacol. 29, 23–36 (1999)
35. Nguyen-Ba, G., Vasseur, P.: Epigenetic events during the process of cell transformation induced by carcinogens (review). Oncol. Rep. 6, 925–932 (1999)
36. Silva Lima, B., Van der Laan, J.W.: Mechanisms of nongenotoxic carcinogenesis and assessment of the human hazard. Regul. Toxicol. Pharmacol. 32, 135–143 (2000)
37. Williams, G.M., Iatropoulos, M.J., Weisburger, J.H.: Chemical carcinogen mechanisms of action and implications for testing methodology. Exp. Toxicol. Pathol. 48, 101–111 (1996)

A Structure Based Algorithm for Improving Motifs Prediction

Sudipta Pathak[1], Vamsi Krishna Kundeti[2], Martin R. Schiller[3], and Sanguthevar Rajasekaran[1,⋆]

[1] Department of Computer Science, University of Connecticut
{sup11002,rajasek}@engr.uconn.edu
[2] Intel Corporation
vamsi.k.kundeti@intel.com
[3] School of Life Sciences, University of Nevada Las Vegas
martin.schiller@unlv.edu

Abstract. Minimotifs are short contiguous peptide sequences in proteins that are known to have functions. There are many repositories for experimentally validated minimotifs. MnM is one of them. Predicting minimotifs (in unknown sequences) is a challenging and interesting problem in biology. Minimotifs stored in the MnM database range in length from 5 to 15. Any algorithm for predicting minimotifs in an unknown query sequence is likely to have many false positives owing to the short lengths of the motifs looked for. Our team has developed a series of algorithms (called *filters*) in the past to reduce the false positives and improve the prediction accuracy. All of these algorithms are based on sequence information. In a recent paper we have demonstrated the power of structural information in characterizing motifs. In this paper we present an algorithm that exploits structural information for reducing false positives in motifs prediction. We test the validity of our algorithm using the minimotifs stored in the MnM database. MnM is a web system for minimotif search that our team has built. It houses more than 300,000 minimotifs. Our new algorithm is a learning algorithm that will be trained in the first phase and in the second phase its accuracy will be measured. For any input query protein sequence, MnM identifies a list of putative minimotifs in the query sequence. We currently employ a series of sequence based algorithms to reduce the false positives in the predictions of MnM. For every minimotif stored in MnM, we also store a number of attributes pertinent to the motif. One such attribute is the *source* of the minimotif. The source is nothing but the protein in which the minimotif is present. For the analysis of our new algorithm we only employ those minimtofis that have multiple sources for positive control. Random data is used as negative data. The basic idea of our algorithm is the hypothesis that a putative minimotif is likely to be valid if its structure in the query sequence is very similar to its structure in its source protein. Another important feature of our algorithm is that it is specific to individual minimotifs. In other words, a unique set of parameters is learnt for every minimotif. We feel that this is a better approach than learning a common set of parameters for all the minimotifs together. Our findings reveal that in most of the cases the occurrences of the minimotifs in their source proteins are structurally similar. Also, typically,

⋆ Corresponding author.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 242–252, 2013.

the occurrences of a minimotif in its source protein and a random protein are dissimilar. Our experimental results show that the parameters learnt by our algorithm can significantly reduce false positives.

1 Introduction

Genetic linkage analysis and other approaches have identified many mutations that are associated with inherited human disease. Many of these mutations are in protein coding regions. An effective strategy for treating many diseases is to identify a drug that interferes with the protein that contains the mutation. Thus, it is important to understand the function of the protein such that drugs can be designed to interfere with its function. Analysis of protein and DNA sequence is an important approach for predicting protein function, thus an important part of the pipeline in drug discovery.

Analysis of DNA and protein sequences often involves the identification of patterns. As a new tool for predicting new causes of disease, our group has built and operates the Minimotif Miner (MnM) website/database (Balla, et al. 2006, Rajasekaran, et al. 2009). MnM can be used to predict potential minimotifs and thus new functions in proteins. These are not domain motifs, but the short functional motif determinants for binding other molecules, the signatures for regulatory posttranslational modifications on proteins, and the short sequence elements that code for protein trafficking. These motifs are readily cross-mapped with disease-associated single nucleotide polymorphisms (SNPs) on the MnM website, thus any scientist can determine a motif that is introduced or eliminated by a disease-associated mutation. One of the principle problems with this approach is that the short motifs are not very complex and false-positives overwhelm the true motifs. In fact all the motif search systems currently available (such as ELM [12], Scansite [7], Prosite [13], Dilimot [14], etc.) suffer from this problem. If this approach were refined, then the approach may be very useful for identifying new drug targets.

In our previous work we have proposed a series of algorithms (called *filters*) (see e.g., [8,9]) to reduce false positives. Examples include protein-protein interaction filter [8], molecular function filter [9], cell function filter [9], etc. These algorithms are all based on sequence information. As is well known, in addition to sequences, structures also contain a rich amount of useful information. In this paper we propose an algorithm for reducing false positives in the prediction of minimotifs. We have tested the accuracy of this algorithm using the minimotifs in MnM. Our empirical tests indicate that the new algorithm is very effective. An interesting feature of our algorithm is that its predictions are specific to individual motifs.

The rest of this paper is organized as follows. In the next section we provide some preliminaries on protein structures. Followed by this we describe our algorithm. Subsequently we provide the results and discussions.

1.1 Some Preliminaries

Every Protein has its primary and secondary structures. Primary structure of a protein is its sequence. The secondary structure consists of helices, sheets, etc. Some of the proteins might have quaternary structures. Protein architecture is one of the most fundamental research topics because the 3D protein structure is responsible for the cell functional properties in all living systems. Amino acid residues are the building blocks of protein primary structure.

The secondary structure of a protein mainly contains the following information: Helix, Sheet, Connectivity Details (disulfide bonds, prolines and other peptides found in cis conformations, etc.), Crystallographic and Coordinate Transformation information (transformation from orthogonal coordinates, transformations expressing non-crystallographic symmetry, etc.), Coordinate Information (collection of atomic coordinates), etc. There exist databases that contain the above information for a subset of the known proteins. An example is the World Wide Protein Data Bank [2]. PIR [15], developed by National Biomedical Research Foundation (NBRF), is one of the earliest primary protein databases. Later in 1988 Martinsried Institute for Protein sequences collected the protein sequences from PIR and developed a web server. Swiss-prot [3] is one of the well known primary protein databases maintained collaboratively by Swiss Institute of Bioinformatics(SIB) and European Bioinformatics Institute(EBI)/European Molecular Biology Laboratory(EMBL). Swiss-prot provides a lot of information including functions of proteins, structures of their domains, post-translational modifications information, etc. This database is a valuable resource produced by PIR from sequences extracted from the Brookhaven Protein Data Bank (PDB). The significance of this database is that it makes available the protein sequence information in the PDB for keyword interrogation and for similarity searches. It includes bibliographic references, MEDLINE cross-references active site, secondary structure and binding site annotations. Also there are composite databases like Non-Redundant DataBase (NRDB)by NCBI (National Center for Biotechnology Information)[5], BLAST (Basic Local Alignment Search Tool) service[16], OWL from the UK EMBnet National Node and the UCL Specialist Node[6] etc. Secondary databases are a consequence of analysis of the sequences of the primary databases, mainly based from Swiss-prot. Prosite [13] is the first among all the secondary databases. This consists of entries about protein families, domains, functional sites, amino acid patterns, etc. This was introduced by Swiss Institute of Bioinformatics and this is mainly based on Swiss-prot.

Along with the above databases a number of web based tools have been developed to allow investigators to search for motifs in a protein query sequence. Scansite [7] is one such tool which includes ten different programs. The Motif Scan ensemble of programs computationally identifies all motifs within a given user-specified protein, while the Database Search ensemble of programs finds all proteins in a protein database, such as Swiss-prot, that match a given motif. One of the most successful tools in this area of research is Minimotif Miner (MnM) that our team has built [8,9,10,11]. All of the known motif search tools suffer from a high false positive rate especially when the motif length is small. We offer a novel solution to this problem in this paper that utilizes structural information.

1.2 Implementation of the Algorithm

To implement the algorithm we make use of Worldwide Protein Data Bank (wwPDB). PDB contains more than eighty thousand proteins and their structural information. We downloaded the entire PDB from the following link: `ftp://ftp.wwpdb.org/pub/pdb/data/structures/divided/`. A typical PDB file contains thousands of lines like the ones shown in Figure 1.

```
.
.
AUTHOR    F.CORDIER,M.S.CAFFREY,B.BRUTSCHER,M.A.CUSANOVICH,D.MARION,
AUTHOR   2 M.BLACKLEDGE
.
.
.
SEQRES   1 A  137  MET LYS ILE SER LEU THR ALA ALA THR VAL ALA ALA LEU
SEQRES   2 A  137  VAL LEU ALA ALA PRO ALA PHE ALA GLY ASP ALA ALA LYS
SEQRES   3 A  137  GLY GLU LYS GLU PHE ASN LYS CYS LYS THR CYS HIS SER
.
.
.
ATOM      1  N   GLY A   1      -7.838 -11.030  -9.480  1.00  0.00           N
ATOM      2  CA  GLY A   1      -7.971 -11.573  -8.138  1.00  0.00           C
ATOM      3  C   GLY A   1      -9.385 -11.361  -7.589  1.00  0.00           C
ATOM      4  O   GLY A   1     -10.177 -10.579  -8.120  1.00  0.00           O
ATOM      5  HA2 GLY A   1      -7.760 -12.642  -8.171  1.00  0.00           H
ATOM      6  HA3 GLY A   1      -7.249 -11.097  -7.476  1.00  0.00           H
ATOM      7  N   ASP A   2      -9.704 -12.054  -6.494  1.00  0.00           N
ATOM      8  CA  ASP A   2     -11.025 -11.993  -5.889  1.00  0.00           C
ATOM      9  C   ASP A   2     -11.203 -10.671  -5.141  1.00  0.00           C
ATOM     10  O   ASP A   2     -11.028 -10.595  -3.927  1.00  0.00           O
ATOM     11  CB  ASP A   2     -11.251 -13.220  -4.999  1.00  0.00           C
ATOM     12  CG  ASP A   2     -12.706 -13.346  -4.553  1.00  0.00           C
ATOM     13  OD1 ASP A   2     -13.311 -12.299  -4.233  1.00  0.00           O
ATOM     14  OD2 ASP A   2     -13.151 -14.504  -4.417  1.00  0.00           O
ATOM     15  H   ASP A   2      -9.000 -12.625  -6.054  1.00  0.00           H
ATOM     16  HA  ASP A   2     -11.772 -12.045  -6.684  1.00  0.00           H
ATOM     17  HB2 ASP A   2     -11.000 -14.117  -5.568  1.00  0.00           H
ATOM     18  HB3 ASP A   2     -10.602 -13.173  -4.127  1.00  0.00           H
.
.
```

Fig. 1. PDB Format

Figure 1 displays the information for the structure of 1C2N. The HEADER, TITLE and AUTHORS records provide information about the investigators involved in defining the structure and other information on the file. The SEQRES records provide the sequences of the peptide chains. We are interested in the ATOM records. The first amino acid GLY (Glycine, symbol G) spans 7 atoms (lines 1-7) and the rest of the atoms correspond to amino acid ASP (Aspartic Acid, symbol D). The 3rd column in each line indicates the type of the atom and the C-alpha atom is indicated by CA (highlighted). The columns 7, 8, and 9 indicate the (X,Y,Z) coordinates of the atom. In the example

of Figure 1, the CA atoms have the coordinates (-7.971, -11.573, -8.138) and (-11.025, -11.993, -5.889), respectively.

The Minimotif Miner (MnM) database contains more than three hundred thousand motifs. We only employ those motifs with multiple sources. Let M_i be such a minimotif that occurs in the following set of source proteins: $S_i = \{s_1, s_2, s_3, \ldots, s_n\}$. Note that, if some motif M_i occurs as a substring in some protein s_j it does not mean that s_j is a source of M_i. Whether this is the case or not can only be experimentally validated. On the contrary, M_i may occur multiple times in its source protein s_j. It is not mandatory that all of these occurrences of M_i in s_j are motifs. At least one of these occurrences of M_i is a motif. So it is not enough for us to know only the source protein ID for a motif. We have to know the location l_k of motif M_i in source s_j. The MnM database provides all such information.

PDB is a much smaller and a slowly growing database than Swissprot/Uniprot. This means that there are many motifs in MnM for which we do not have a valid PDB ID. MnM uses a variety of IDs for proteins including Uniprot/Swissprot and Refseq. The mapping between MnM and PDB is done using the mapping files obtained from the following link : `http://www.bioinf.org.uk/pdbsprotec/mapping.txt`.

We have implemented our algorithm using the Center of Gravity algorithm for computing the distance between two structures [1]. The Center of Gravity algorithm is described in the next subsection.

1.3 Center of Gravity Algorithm

This algorithm can be applied to compute the distance between two point sets in any n-dimensional Euclidian space. We explain the algorithm for 3-dimensional case because of simplicity and the scope of our work.

Input : This algorithm takes as input two sets of (x,y,z) coordinates. These are given by $S^{(x,y,z)_i} = \{(x_1^i, y_1^i, z_1^i), (x_2^i, y_2^i, z_2^i), \ldots, (x_n^i, y_n^i, z_n^i)\}$ and $S^{(x,y,z)_j} = \{(x_1^j, y_1^j, z_1^j), (x_2^j, y_2^j, z_2^j), \ldots, (x_n^j, y_n^j, z_n^j)\}$.

Output: Distance between $S^{(x,y,z)_i}$ and $S^{(x,y,z)_j}$. We call it CoG distance.

Algorithm:

BEGIN
Compute (x,y,z) coordinates of the centroid of $S^{(x,y,z)_i}$.
This is given by (x_c^i, y_c^i, z_c^i);
Compute (x,y,z) coordinates of the centroid of $S^{(x,y,z)_j}$.
This is given by (x_c^j, y_c^j, z_c^j);
for each of the coordinates $(x_q^i, y_q^i, z_q^i) \in S^{(x,y,z)_i}$ do
  compute the Euclidian distance between (x_c^i, y_c^i, z_c^i) and (x_q^i, y_q^i, z_q^i).
Let the set of distances be given by $D_i^{Euclidian} = \{d_1^i, d_2^i, \ldots, d_n^i\}$;
for each of the coordinates $(x_q^j, y_q^j, z_q^j) \in S^{(x,y,z)_j}$ do
  compute the Euclidian distance between (x_c^j, y_c^j, z_c^j) and (x_q^j, y_q^j, z_q^j).
Let the set of distances be given by $D_j^{Euclidian} = \{d_1^j, d_2^j, \ldots\ldots, d_n^j\}$;
CoG distance is given by $D_{ij}^{CoG} = \sqrt{(d_1^i - d_1^j)^2 + (d_2^i - d_2^j)^2 + \ldots + (d_n^i - d_n^j)^2}$.
END

2 Methods

Our algorithm is based on the following hypothesis: Positive occurrences of the same motif in different sources are structurally similar. Also, the structure of a positive occurrence of a motif and any of its negative occurrences will be dissimilar. To compute the distance between two structures we employ the center of gravity algorithm proposed in [1].

Our algorithm is a learning algorithm that has to be trained with a set of positive and negative examples in the first phase. We evaluate its accuracy in the second phase. A special feature of our algorithm is that it learns the relevant parameters for each individual motif separately. It turns out there is only one parameter that is learnt. This parameter is nothing but a distance threshold between two structures. Let M be any motif. If O_1 and O_2 are the structures corresponding to two positive occurrences of M, then we expect the distance between O_1 and O_2 to be 'small'. On the other hand, if O_1 corresponds to a positive occurrence and O_2 corresponds to a negative occurrence, then we expect the distance between them to be 'large'. Since any learning algorithm requires multiple positive and negative examples to learn from, and our algorithm is motif-specific, we only employ those validated minimotifs in MnM that have multiple sources. Each such source serves as a positive example. Finding negative examples for any biological experiment is in general a challenge since we may not be able to be sure that any data is negative. Like in our previous works on filters, in this paper also we employ random data as negative data. As has been argued before, a random data has a very high probability of being negative.

If M is a motif under concern and if its known sources are $S_1, S_2, \ldots, S_n$, we first get all the occurrences of M in each of the sources. Let these occurrences be $O_1, O_2, \ldots, O_m$. Our hypothesis states that the O_is are structurally similar. Since a motif can occur more than once in the same source, it is the case that $m \geq n$. By structure information we mean a point set in 3D. Specifically, by structure we mean the set of coordinates of the alpha carbon atoms in the motif sequence. This information is available in the PDB files. In this paper we consider only the alpha carbon atoms. Note that including other atoms would only improve the prediction accuracy further. In the final version of the paper we will include other atoms as well.

2.1 Steps in the Algorithm

1. Get a list of all the validated motifs in the MnM database that have multiple sources.
2. Let M be any motif whose sources are $S_1, S_2, \ldots, S_q$. For these source proteins Refseq IDs are available in MnM.
3. We keep only those sources for which structure information is available in PDB. This is done using a Refseq ID$\rightarrow$ PDB ID mapping table.
4. For a given motif M, let its sources for which we are able to get PDB IDs be $S_1, S_2, \ldots, S_n$. We pick one of these sources as the reference for our experiment and call it S_{ref}. The others are used as positive controls. In other words, they serve as positive examples in learning.
5. For each of the positive controls and the reference we apply the Center of Gravity algorithm to perform the following tasks:

   a. Compute the Center of Gravity of the alpha carbon atoms in the motif sequence.
   b. Compute the Euclidean distances between each of the alpha carbon atoms and the center of gravity. Let these distances in sorted order be $d_1, d_2, \ldots, d_l$, where l is the length of the motif. Note that for every amino acid in the motif there is a single alpha carbon atom. Also note that we will get one such sorted set $\{d_1, d_2, \ldots, d_l\}$ for each of the positive controls.
   c. Let the set of distances for the reference S_{ref} be given by $\{d_1^{ref}, d_2^{ref}, \ldots, d_l^{ref}\}$.
   d. Calculate the Euclidean distance between $\{d_1^{ref}, d_2^{ref}, \ldots, d_l^{ref}\}$ and $\{d_1, d_2, \ldots, d_l\}$ for each positive control. Let the Euclidean distance for the jth positive control be d_j.
   e. Take an average over all the d_js. This is called the *positive mean*.
6. For a given motif M scan through the PDB to look for proteins which are not known to be source proteins for M and in which M occurs as a substring. In other words, exclude the set of positive controls and the reference from the set of all proteins in PDB where M occurs as a substring. This new set is used as the set of negative controls for the motif M. Let this set be $\{N_1, N_2, \ldots, N_t\}$.
7. For each of these negative controls and the reference protein we again apply the Center of Gravity algorithm and compute a distance as in step 5. This will give us the Euclidean distance between $\{d_1^{ref}, d_2^{ref}, \ldots, d_l^{ref}\}$ and $\{d_1, d_2, \ldots, d_l\}$ for each negative control. Let the Euclidean distance for the kth negative control be d_k. We get an average over all of these d_ks and obtain the *negative mean*.
8. We have to come up with a threshold using which we can separate the true positives and false positives. One possibility is to use the negative mean as the threshold. In this case we compute how many of the positive distances d_js are above the negative mean and how many of the negative distances d_ks are above the negative mean.

We expect that a large fraction of positive control distances will be below the negative mean based on our hypothesis.

3 Results

We have tested our algorithm on a collection of almost 650 motifs (that have multiple sources). We have performed two types of analyses. The first analysis is to test the statistical significance of the results obtained using ROC plots. The second analysis measures the accuracy of predictions.

3.1 ROC Plots

For each motif M_i we compute its negative mean $D^-_{M_i}$ and use it as a threshold for predictions. We calculate the number $Count^+_{M_i}$ of distance values below the threshold value, from among the true positive occurrences. This count gives us the true positive rate (TPR). We also calculate the number $Count^-_{M_i}$ of distance values below the same threshold value from among the false positives (i.e., negative control). This number will give us the false positive rate (FPR). According to our hypothesis there should be a good structural similarity between occurrences of a motif in its source proteins. This

means that the CoG distance between any two occurrences of the motif in its sources is supposed to be smaller compared to the CoG distance of a true positive occurrence of the motif and a false occurrence of the same motif. We plot FPR (as horizontal axis) vs TPR (as vertical axis) curve and calculate the area under the curve (AUC) for various threshold values. We do this for all the 650 motifs. Table 1 summarizes the outcomes of our experiment.

Table 1. Areas Under the Curves

$AreaUnderCurve(AUC)$	Number of Motifs (as a %)
$> 90\% and \leq 100\%$	29.629
$> 80\% and \leq 90\%$	7.407
$> 70\% and \leq 80\%$	9.259
$> 60\% and \leq 70\%$	7.407
$> 50\% and \leq 60\%$	18.518
$< 50\%$	27.777

Out of the 650 motifs (each having 67.85 positive controls on an average) used for analysis, 216 have got an area under the curve (AUC) between 0.9 and 1. For almost 58 motifs the AUC is exactly 1. This demonstrates the power of our algorithm. The idea is to use our new algorithm only for those motifs for which the AUC is at a level comfortable to a biologist.

3.2 Accuracy Calculation

Accuracy is defined in the following equation:

$$Accuracy = \frac{Number\ of\ +ve\ distances\ below\ threshold\ +\ Number\ of\ -ve\ distances\ above\ threshold}{Total\ number\ of\ distances}$$

Table 2 shows number of motifs in different intervals of accuracy.

We have almost 147 motifs with a prediction accuracy between 90% and 100%. Here again the filter corresponding to the new algorithm is to be used for only those motifs for which the accuracy is at an acceptable level. Figure 2 displays the ROC plots for a randomly chosen subset of the motifs. We show two ROC plots for each category of Table 2.

Table 2. Accuracy

Accuracy	Number of Motifs (as a %)
$> 90\% and \leq 100\%$	22.727
$> 80\% and \leq 90\%$	9.09
$> 70\% and \leq 80\%$	19.696
$> 60\% and \leq 70\%$	15.151
$> 50\% and \leq 60\%$	33.333
$< 50\%$	0

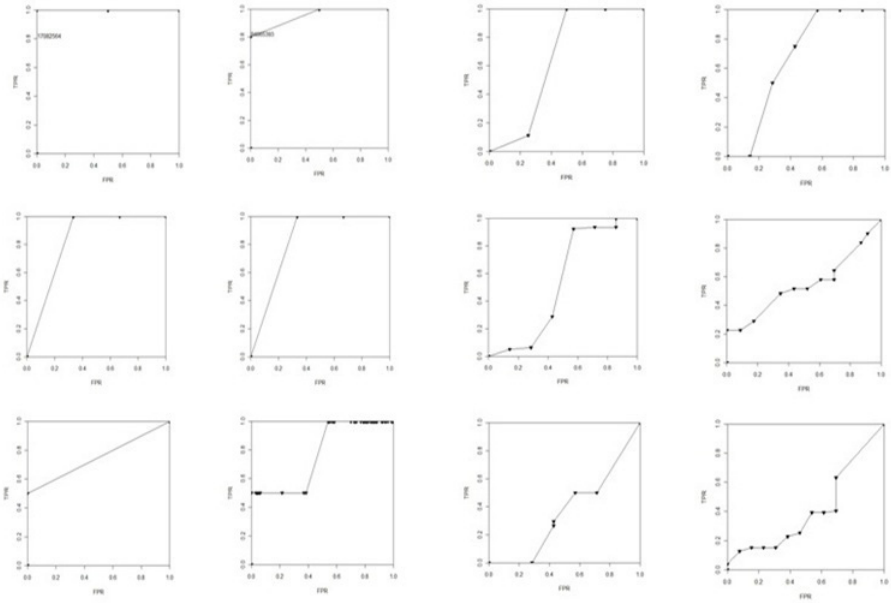


Fig. 2. ROC plots

We plan to integrate the entire data and code as a part of the MnM web system. We will associate a threshold and accuracy/AUC with each of the motifs in the MnM database. Once a user enters a protein query Q, MnM reports the putative motifs in Q. For any motif M if the query is one of the known sources then M is reported as a true prediction with an accuracy of 100%. One the contrary, if Q is not one of the known

sources of M, the filter checks to see if Q is present in PDB. If Q is found in PDB we apply the center of gravity algorithm to compute the CoG distance D_M^Q for M in Q. If the difference between D_M^Q and the CoG distance of M in the reference protein is below the threshold set for M in the MnM database, then M is reported to be a true motif. Accuracy of prediction and AUC value is also reported by MnM. If D_M^Q is above the threshold we will not report M as a putative motif.

4 Conclusion and Future Work

In this paper we have presented a novel structure based algorithm for reducing false positives in the prediction of minimotifs. Our algorithm is a motif-specific learner. We live in an era of personalized medicine and hence this approach is very relevant. The statistical significance of the results obtained as well as the accuracy of the new algorithm demonstrate that the new algorithm is indeed very effective. The outcomes of this work points to the following directions for future work. We want to consider the coordinate information of all the atoms in the amino acids. We want to see the best possible set of features to come up with a better classification accuracy. As mentioned earlier this could only improve the result. Also, we choose the positive reference arbitrarily. We want to extend our the work by choosing each of the positive instances as a possible reference. We will calculate the area under curve and accuracy for each one of them. Finally we choose the best of these scores and the reference associated with it.

Acknowledgements. This work has been supported in part by the following grants: NSF 0829916 and NIH R01-LM010101. We also thank David Sargent for many fruitful discussions.

References

1. Kundeti, V.K., Rajasekaran, S.: A Statistical Technique to Predict Structural Characteristics of Short Motifs, BECAT Tech. Report
2. Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description Version 3.30 Document Published by the wwPDB
3. UniProt Documentation, `http://www.ebi.ac.uk/uniprot/Documentation/`
4. Database of protein domains, families and functional sites, `http://prosite.expasy.org/prosite.html/`
5. Non-redundant databases (NRDB)
6. OWL database, `http://www.bioinf.man.ac.uk/dbbrowser/OWL/index.php`
7. Obenauer, J.C., Cantley, L.C., Yaffe, M.B.: Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic Acids Research 31(13), 3635–3641 (2003)
8. Rajasekaran, S., Merlin, J.C., Kundeti, V., Oommen, A., Mi, T., Oommen, A., Vyas, J., Alaniz, I., Chung, K., Chowdhury, F., Deverasatty, S., Irvey, T.M., Lacambacal, D., Lara, D., Panchangam, S., Rathnayake, V., Watts, P., Schiller, M.R.: A computational tool for identifying minimotifs in protein-protein interactions and improving the accuracy of minimotif predictions. Proteins: Structure, Function, and Bioinformatics 79(1), 153–164 (2010)

9. Rajasekaran, S., Mi, T., Merlin, J.C., Oommen, A., Gradie, P., Schiller, M.R.: Partitioning of minimotifs based on function with improved prediction accuracy. PLoS ONE 5(8), e12276 (2010)
10. Rajasekaran, S., Balla, S., Gradie, P., Gryk, M.R., Kadaveru, K., Kundeti, V., Maciejewski, M.W., Mi, T., Rubino, N., Vyas, J., Schiller, M.R.: Minimotif miner 2nd release: a database and web system for motif search. Nucleic Acids Research 37, D185–D190 (2009)
11. Balla, S., Thapar, V., Verma, S., Luong, T., Faghri, T., Huang, C.-H., Rajasekaran, S., del Campo, J.J., Shinn, J.H., Mohler, W.A., Maciejewski, M.W., Gryk, M.R., Piccirillo, B., Schiller, S.R., Schiller, M.R.: Minimotif Miner, a tool for investigating protein function. Nat. Methods 3, 175–177 (2006) (PMID: 16489333)
12. Via, A., Gould, C.M., Gemünd, C., Gibson, T.J., Helmer-Citterich, M.: A structure filter for the Eukaryotic Linear Motif Resource. BMC Bioinformatics 10, 351 (2009), doi:10.1186/1471-2105-10-351
13. Sigrist, C.J.A., Cerutti, L., Hulo, N., Gattiker, A., Falquet, L., Pagni, M., Bairoch, A., Bucher, P.: PROSITE: A documented database using patterns and profiles as motif descriptors. Oxford Journals (2002), doi: 10.1093/bib/3.3.265
14. Neduva, V., Russell, R.B.: DILIMOT: discovery of linear motifs in proteins. Nucleic Acids Res. (2006), doi: 10.1093/nar/gkl159
15. Sidman, K.E., George, D.G., Barker, W.C., Hunt, L.T.: The protein identification resource (PIR). Nucleic Acids Research 16(5) (1988)
16. Altschul, S.F., Gish, W., Myers, W.M.E.W., Lipmanl, D.J.: Basic Local Alignment Search Tool. J. Mol. Biol. 215, 403–410 (1990)

A Workflow for the Prediction of the Effects of Residue Substitution on Protein Stability

Ruben Acuña[1], Zoé Lacroix,[1], and Jacques Chomilier[2,3]

[1] Arizona State University, Tempe, AZ 85287, USA
[2] Protein Structure Prediction, IMPMC, Université Pierre et Marie Curie, CNRS UMR 7590, Paris, France
[3] RPBS, Université Paris Diderot, 35 Rue Hélène Brion, Paris, France

Abstract. The effects of residue substitution in protein can be dramatic and predicting its impact may benefit scientists greatly. Like in many scientific domains there are various methods and tools available to address the potential impact of a mutation on the structure of a protein. The identification of these methods, their availability, the time needed to gain enough familiarity with them and their interface, and the difficulty of integrating their results in a global view where all view points can be visualized often limit their use. In this paper, we present the Structural Prediction for pRotein fOlding UTility System (SPROUTS) workflow and describe our method for designing, documenting, and maintaining the workflow. The focus of the workflow is the thermodynamic contribution to stability, which can be considered as acceptable for small proteins. It compiles the predictions from various sources calculating the $\Delta\Delta$G upon point mutation, together with a consensus from eight distinct algorithms, with a prediction of the mean number of interacting residues during the process of folding, and a sub domain structural analysis into fragments that may potentially be considered as autonomous folding units, i.e., with similar conformations alone and in the protein body. The workflow is implemented and available online. We illustrate its use with the analysis of the engrailed homeodomain (PDB code 1enh).

1 Introduction

As it has been reviewed by Tokuriki and Stawfik [42], amino acid substitution is now considered as a major constraint on protein evolvability, while it was previously admitted that most positions can tolerate drastic sequence changes, provided the fold is conserved. Actually, mutations affect stability and stability affects evolution. The level of deleterious mutations can be as high as one third [42]. Therefore, the prediction of the effects of residue substitution can be of great help in wet labs. In this paper, we focus only on the thermodynamic contribution to stability, which can be considered as acceptable for small proteins.

Potapov et al. [37] compared six different methods to predict the change in protein stability on a set of mutations taken from the FOLDEF paper [13] and a

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 253–264, 2013.

second set from ProTherm [17]. The tested tools are: CC/PBSA [3], EGAD [35], FoldX [41], Hunter [36], Imutant2 [4] and Rosetta [39]. The authors notice that Rosetta is not trained for $\Delta\Delta$G calculations, thus resulting in a low correlation coefficient compared to EGAD, the best in their study. One of the drawbacks of EGAD is the fact that they do not predict special mutations, namely Cys, Gly and Pro, because the perturbation to the backbone is too large with these residues. One can nevertheless notice that none of the methods is able to correctly predict all the $\Delta\Delta$Gs for all mutations, but the general trend is correct on average. The average error is 1.72 kcal/mol, thus one can reasonably put a threshold at 2 kcal/mol for the decision of hot spot positions. Khan and Vihinen [15] completed another study with: Automute [28], Cupsat [34], Dmutant [50], FoldX, Multimutate [8], Mupro [5], Imutant versions 2 and 3 [4], and the set SCide [9], SCpred [18] and SRide [27]. The latter three programs identify stability centers rather than provide a general prediction of $\Delta\Delta$G and so were excluded from our selection. Khan and Vihinen also examined Automute [28] but could not produce enough test data for statistical analysis.

Scientists interested in the prediction of the effects of residue mutations have therefore to select a tool among many tools available to compute such prediction [3,35,41,36,4,39,28,34,50,8,5,50,8,4,9,18,27], get familiar with its interface and various built-in specifications and limitations (not always documented), run the execution of the selected tool, and compare, often manually, the results with results obtained with a similar tool or a tool implementing a complementary concept. A benchmark study such as [37] may guide the selection of a tool, in contrast we demonstrate the benefits of a workflow that orchestrates the best tools, integrates the results and compiles a consensus into a single interface.

Workflows are used in business applications to assess, analyze, model, define and implement business processes. A workflow automates the business procedures where documents, information or tasks are passed between participants according to a defined set of rules to support an overall goal. In the context of scientific applications, a workflow approach may promote collaboration among scientists, as well as the integration of scientific data and tools. Scientific workflows focus on the support of scientific experiments replay, design and data retrieval whereas Laboratory Information Management Systems (LIMS) support the integration of different functionalities in a laboratory, such as sample tracking (invoicing/quoting), integrated bar-coding, instrument integration, personnel and equipment management, etc.

The Structural Prediction for pRotein fOlding UTility System (SPROUTS) workflow was developed to provide a global view of the potential impact of mutations on proteins. It aims at integrating several concepts and implements each of them with various methods and tools. In this paper we focus on the predictions from eight resources calculating the $\Delta\Delta$G upon point mutation and a consensus method.

The paper is organized as follows. We first discuss work related to scientific workflows in Section 2. The development of a scientific workflow requires addressing many challenges including design, implementation, maintenance, and

performance. They are discussed in the context of the SPROUTS workflow in Section 3 whereas our approach is described in Section 4. We present a use case in Section 5. Future work is presented in Section 6.

2 Scientific Workflows

Scientific workflows often are executed manually. The reasons for manual executions include, among others, the need to validate the results of intermediate steps, the benefit of graphical interfaces of the tools they integrate, the better knowledge of the resource functionalities by experiencing them manually, the changes and updates made on resources that are more easily traceable when the user is using them. These processes are very often poorly documented and scientists experience difficulties in reproducing their datasets as the resources they use may change over time (new database entries, data curation, new version of a tool, etc.). This lack of documentation also affects the ability of integrating and comparing datasets and analyses produced over time. Moreover, the manual execution of a workflow is typically time and manpower consuming. Scripting programming environments such as Perl and Python have also been proven incredibly successful to support the rapid development of workflows. Although this automation saves time and manpower they typically fail to support the proper design and documentation of the process. Lack of documentation not only affects data integration and comparison but also workflow re-use and revision. Various Web-based work benches offer an alternative solution to the problem of automation of orchestrated bioinformatics resources by providing unified access with a simplified interface to multiple resources running on their servers. They include PISE [23], wEMBOSS [40], and Mobyle [33] among many others.

Workflow systems are very successful among the biological community as they provide scientists with the ability to express their scientific protocols as a sequence of connected steps [22]. They describe the scientific process from experiment design, data capture, integration, processing, and analysis that leads to scientific discovery. They typically express *digital* workflows and execute them on platforms such as grids. The procedural support of a workflow resembles the query-driven design of scientific problems and facilitates the expression of scientific pipelines (as opposed to a database query). Kepler [25], which extends the Ptolemy II system [29,30], supports modular workflow design and task scheduling. WOODSS [31] emphasize the support of several abstraction levels of workflow design and facilitates workflow composition and reuse. Many scientific workflow systems focus on execution in general [29,1] or in the Grid computing environment. For example, the GriPhyN Project [12] is developing Grid technologies to collect and analyze distributed scientific and engineering datasets. The Pegasus framework [7,26] uses the Chimera system [10] to describe abstract workflows, and Condor DAGMan and schedulers [6] to generate concrete workflows for execution on the Grid. In Taverna [14] a workflow is composed of *processors* connected with data dependencies links. Its revised updated version is now extensible and scalable that can be used from a workbench, a command

line or remotely as a server [32]. One of the challenges not yet addressed by these approaches is the legacy of scientific workflows. Indeed while they offer support for the development of new workflows the automation, documentation, and revision of legacy workflows such as SPROUTS remains a challenge.

3 The SPROUTS Workflow

The initial process was designed to populate the SPROUTS database [24] with six tools: DFIRE version 2.0 [49], I-Mutant 2.04 and I-Mutant-DSSP 2.04 [4], MUpro version 1.1 [5], PoPMuSiC [11], and a stability consensus method. The development of the new revised workflow followed three successive revision steps: automation of the database population process, update of the workflow with more recent tools, and support of on-line submission of proteins.

To compose the revised SPROUTS workflow we concentrated on programs that could be run on our servers, therefore excluding the Web submission systems, such as Eris [48] (the standalone version is commercial), Cupsat, Automute, in order to avoid manipulation of various formats when new releases of the programs are proposed. We had intended to include CUPSAT, however, we were unable to contact the authors due to issues with their website and contact addresses. We performed trial use of MultiMutate but found it incompatible (unstable) with the existing (Ubuntu based) server that the workflow must execute on. In addition to the tools analyzed by Khan and Vihinen, we also examined SDM [46] and Pro-Maya [44] but they are not currently available as a local executable or a Web service and so cannot be integrated with the existing workflow.

The new revised SPROUTS workflow processes data for DFIRE 1.1 (Dmutant), FoldX 3.0 beta 5.1, I-Mutant 2.0 sequence/structure modes, I-Mutant 3.0 sequence/structure modes, and MUpro. Our database also contains legacy data from PoPMuSiC [19], these data were part of the original database. Because no local executable version of this tool was available, we were unable to include it in our workflow. These represent the most recent versions of the respective tools with one exception: DFIRE 2.1 [47]. This most recent version of DFIRE operates directly on a (possibly) mutated PDB structure. Because our current workflow does not support dependencies between tools, we were unable to produce the necessary mutant PDBs to use DFIRE 2.1. MuD [45] an interactive Web server for the prediction of mutations from a structure based on a machine learning algorithm was published recently. We have not integrated this tool yet because it does not provide a $\Delta\Delta$G calculation but rather an estimate of function conservation. The revision of the SPROUTS workflow with MuD would require changing the consensus method in such a way the $\Delta\Delta$G calculation of the other tools can be combined with an estimate of the function conservation.

The SPROUTS workflow[1] populates the SPROUTS database [24]. Submitting a new protein to the SPROUTS system executes the whole SPROUTS workflow and uploads the results into the database. Because the execution of the workflow

[1] The SPROUTS workflow is available online at `http://bioinformatics.engineering.asu.edu/sprouts`.

takes time, a link to the the database entry is provided to the user to retrieve the results from the database after completion of the workflow. The user retrieves the information pertaining to one protein at a time through its PDB ID. The user may then select a single tool or access the results of all the tools. A specific residue and mutation may also be selected (by default every residue and every mutation will be returned). The residue number may be specified (note that SPROUTS numbering does not follow PDB numbering: in case the user specifies an amino acid and a number, SPROUTS will check if this is the right amino acid at this position). By default, no residue is specified and so all residues will be considered. Another parameter offers the possibility to visualize only the mutations which increase the stability or at the opposite which decrease the stability (the default mode is to return all results. The last parameter offers the possibility to limit the number of results displayed on the result page. By default, the value is set to 190 lines which correspond to the results of all the 19 possible mutations for 2 residues and for all the tools. Even if the option is available, it is strongly advised not to select the "all" option especially for long proteins.

4 Developing the SPROUTS Workflow

Our method for workflow development involves the characterization of the workflow at four levels: semantic, implementation, execution, and data. To document the workflow we follow the approach developed with ProtocolDB where workflows are first expressed in terms of a domain ontology where each task expresses a specific aim [16]. Domain ontologies[2] can be used to describe the concepts and relationships of a discipline as well as to document the tools and methods [20]. A *design protocol* (or workflow) is defined top-down from a conceptual design task that describes the workflow as a whole. The conceptual design is defined in terms of input and output parameters which are expressed as complex conceptual types (collections of concept variables). Each design task may be split either sequentially (with the $\otimes$ operator) or in parallel (with the $\oplus$ operator) into two design tasks. The semantic characterization of the workflow enables reasoning on workflows at a conceptual level. Semantic equivalence of workflow implementations (mapped to the same semantic representation) can be used to validate data integration, compare implementations performances and support workflow optimization [21].

The concepts involved in the SPROUTS workflow include `Protein`, specified with its name and PDB code, sequence, structure, and secondary structure, `Residue`, specified by its name and location on the sequence, and the value of Gibbs free energy as an approximation to characterize the stability of a given structure. See [43] for an ontology devoted to structural bioinformatics. We consider the difference of energy for the wild type of the protein ΔG_{wild} and for the mutant ΔG_{mutant}. We define[3] the difference as $\Delta\Delta G = \Delta G_{mutant} - \Delta G_{wild}$ in

[2] See The Open Biological and Biomedical Ontologies at `http://www.obofoundry.org/` for a list of ontologies for various scientific domains.

[3] Note that different stability prediction methods use different definitions.

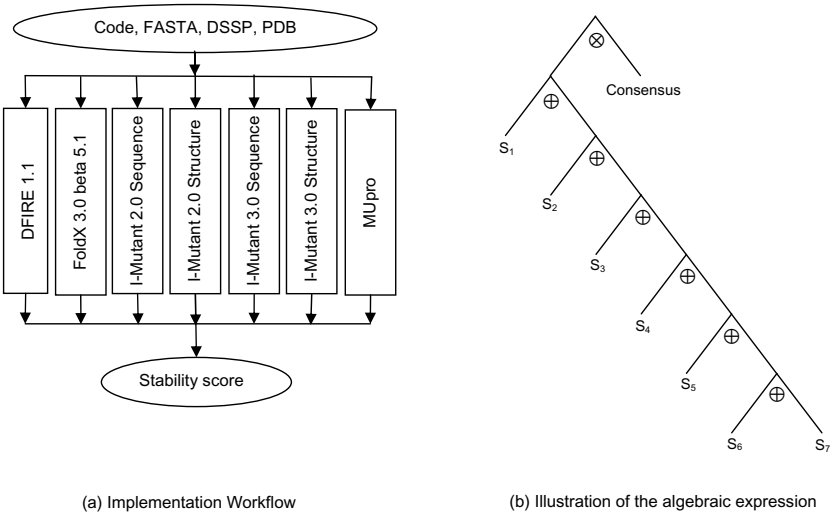


Fig. 1. SPROUTS Implementation Workflow

kcal/mol. At this level of definition, the workflow consists of a single task that links the concept `Protein` to a score that expresses the impact of a mutation on its stability for each residue. The revisions of the workflow discussed in Section 3 do not impact the semantics of SPROUTS. The phase of automating the legacy population workflow does not change its semantics nor does updating the tools the workflow is composed of. Indeed, the *design workflow* captures the semantic aim of the workflow which is not affected by the proposed revision.

The second development phase consists of the specification of the resources that are implementing each of the design tasks. Each design task is mapped to a *implementation protocol* (or workflow) defined as follows. An *implementation protocol* is a graph composed of connected scientific resources (database queries or tools) whose inputs and outputs are data types. A single bioinformatics service is an implementation protocol. Complex implementation protocols are composed of scientific resources connected with the same two binary operators $\oplus$ and $\otimes$ used to express design protocols. Here, the design task can be implemented by many existing resources as discussed in Section 1. Because we chose to exploit multiple stability prediction methods, and integrate their results in a consensus step, the single design task will be first mapped to two successive implementation steps connected with the $\otimes$ operator. The second implementation step will be specified with the consensus method. The first implementation step will be split with the parallel operator $\oplus$. The first of the two steps will be specified with the first stability prediction tool DFIRE 1.1 when the second one will be split into two parallel steps. The first of the two will be assigned to FoldX 3.0 beta 5.1 whereas the the second one will be, again, split into two parallel steps, and so on.

The resulting implementation workflow is expressed by

$$(S_1 \oplus (S_2 \oplus (S_3 \oplus (S_4 \oplus (S_5 \oplus (S_6 \oplus S_7)))))) \otimes Consensus$$

where $S_1, \ldots S_7$ denote respectively DFIRE 1.1, FoldX 3.0 beta 5.1, I-Mutant 2.0 sequence, I-Mutant 2.0 structure, I-Mutant 3.0 sequence, I-Mutant 3.0 structure, and MUpro.

The input (resp. output) of the implementation workflow consists of the input (resp. output) datatype. The input of the implementation workflow describes the concept `Protein` as follows. It consists of a 4-character code (that may be a PDB ID), the protein primary structure or sequence in FASTA format, the description of the secondary structure in DSSP format, and the 3-D structure in PDB format. The output consists of the protein sequence (list of residues) and the stability scores (for each residue, 8 scores are computed: one for each tool and the consensus score). The SPROUTS implementation workflow illustrated in Figure 1 can be represented with a binary tree.

The third level of workflow characterization is the execution plan. This level requires the specification of the programmatic environment (e.g., Taverna, Kepler, or scripting language such as Python, Perl). The first step of the SPROUTS workflow revision (automation of the database population process) did not affect the first two layers of representation. The revision consisted in replacing the manual execution by a Python program. Although the orchestration of the steps that were initially used to populate the database into a single script was not likely to produce a well designed workflow with suitable performance and adaptability, it was the chosen path because it was also the one less likely to impact the availability of the SPROUTS database. The second step of the revision (workflow update with more recent tools) had an impact on both the implementation layer as new tools were used and the execution layer as the overall structure of the workflow had also changed. The main challenges of SPROUTS development have been importing applications and tools which lack documentation, including the specification of the limitations (often implicit) of their input, a description of their computational time (performance) and execution failures. Moreover, none of the tools exploited in the workflow offers a description of its interface expressed in a machine readable format such as Web Service which limits the ability of implementing and executing the workflow on a system such as Taverna.

5 Use Case

The SPROUTS workflow is implemented and available online. Once the protein has been submitted to the SPROUTS workflow and the execution has completed, the results are stored in the SPROUTS database and can be accessed with the query form. All stability prediction tools of the workflow are selected by default.

The results for 1enh are shown in the table (left of Figure 2). In the 2D mode, the results of all the tools but FoldX 3 are displayed (right). The consensus graph is currently created by taking the mean of the available data. Due to evolution,

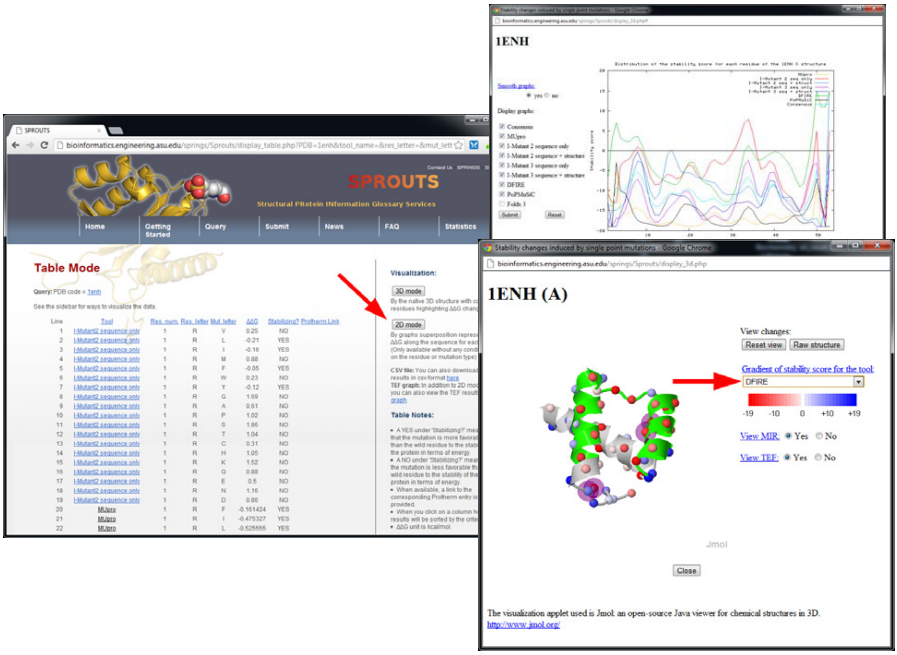


Fig. 2. Engrailed homeodomain (PDB code 1enh)

the number of stabilizing mutations is smaller than destabilizing ones. One must mention that a stabilizing mutation is not necessarily related to an improved efficiency of the mutated protein, as far as function is concerned. Sometimes, a more stable structure results in an increased rigidity, while the function requires a certain level of flexibility. This is the case for instance with enzyme catalysis [46]. Therefore, it seems reasonable to place a threshold of 2 kcal/mol in either way of $\Delta\Delta$G (stabilizing or destabilizing) in order to claim to a putative malfunction. Mutations in conserved positions usually cause large stability decreases. The 3D mode (bottom right) displays the protein structure retrieved from PDB.

The engrailed homeodomain (PDB code 1enh) is a small single domain (54 residues), monomeric, composed of three helices, and without any disulfide bridge. It is considered as a model for the hierarchic type of folding, and one Leucine, at position 14, is deeply buried in the core of the structure, stabilized by hydrophobic interactions with amino acids from the two other helices. This particular residue has been mutated by the group of Fersht [38] and the NMR structure determined (PDB code 1ztr). The mutated form is no more a globular protein, since the accessible surface area is increased by 50% due to mutation. Nevertheless, most of the local stability remains since the three helices are still present.

When comparing the 1enh and 1ztr 2D plots, the differences are not significant, unless some N and Cter effects due to the non symmetrical process of smoothing. But introducing the structure in the algorithm has an effect in

I-Mutant. Although the general shape is similar between 1enh and 1ztr for I-Mutant 2.0 with structure, the highest divergence occurs around position 14. Such a peak does not appear in the two algorithms considering only the sequence. It pleads in favor of the proof of a better prediction with structures included, specially when single mutations are concerned. When comparing now the two versions of I-Mutant (2.0 vs 3.0) the high peak of instability is conserved for 1enh around position 8. But the peak previously discussed around 15 in I-Mutant 2.0 almost vanishes with I-Mutant 3.0. Nevertheless, although the peak decreases in the middle of the first helix, the global gross features of the shape of the curves are looking like for the wild type structure. This is not the case for the mutated structure, and one may argue that the underlying principles ruling I-Mutant 3.0 are scaled on compact globular proteins, and do not apply to proteins looking like NUP (Natively Unfolded Protein).

6 Conclusion and Future Work

The workflow is under significant revision and extension with new functionalities and improved interface to come. Once the revision is completed, a mirror of SPROUTS 2.0 will be deployed in the Ressource Parisienne en Bioinformatique Structurale (RPBS) [2]. The current version of the SPROUTS workflow is available at http://bioinformatics.engineering.asu.edu/springs/Sprouts/.

Acknowledgment. We acknowledge and thank Pierre Tufféry, Dirk Stratmann, Elodie Duprat, Mathieu Lonquety, Christophe Legendre, Nikolaos Papandreou, Fayez Hadji, as well as the authors of the different tools used in SPROUTS. We also wish to acknowledge our collaborators at ASU: Rida Bazzi, Antonia Papandreou-Suppappola, Anna Malin, and Banu Ozkan. This research was partially supported by the National Science Foundation[4] (grant CNS 0849980) and an invitation by the Université Pierre et Marie Curie.

References

1. Aeschlimann, M., Dinda, P., Lopez, J., Lowekamp, B., Kallivokas, L., O'Hallaron, D.: Preliminary report on the design of a framework for distributed visualization. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 1833–1839 (1999)
2. Alland, C., Moreews, F., Boens, D., Carpentier, M., Chiusa, S., Lonquety, M., Renault, N., Wong, Y., Cantalloube, H., Chomilier, J., Hochez, J., Pothier, J., Villoutreix, B.O., Zagury, J.-F., Tufféry, P.: RPBS: a web resource for structural bioinformatics. Nucleic Acids Res. 33(web Server issue), W44–W49 (2005)
3. Benedix, A., Becker, C.M., de Groot, B.L., Caflisch, A., Böckmann, R.A.: Predicting free energy changes using structural ensembles. Nat. Methods 6(1), 3–4 (2009)

[4] Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

4. Capriotti, E., Fariselli, P., Casadio, R.: I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. Nucleic Acids Res. 33, 306–310 (2005)
5. Cheng, J., Randall, A., Baldi, P.: Prediction of protein stability changes for single site mutations using support vector machines. Proteins 62, 1125–1132 (2006)
6. Condor. Manual (7.0.1) (2008), http://www.cs.wisc.edu/condor/manual/v7.0/
7. Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.-H., Vahi, K., Livny, M.: Pegasus: Mapping Scientific Workows onto the Grid. In: European Across Grids Conference, pp. 11–20 (2004)
8. Deutsch, C., Krishnaoorthy, B.: Four body scoring function for mutagenesis. Bioinformatics 23(22), 2009–3015 (2007)
9. Dosztányi, Z., Magyar, C., Tusnády, G., Simon, I.: SCide: identification of stabilization centers in proteins. Bioinformatics 19(7), 899–900 (2003)
10. Foster, I., Voeckler, J., Wilde, M., Zhao, Y.: Chimera: a virtual data system for representing, querying and automating data derivation. In: 14th International Conference on Scientific and Statistical Database Management, pp. 37–46 (2002)
11. Gilis, D., Rooman, M.: PoPMuSiC, an algorithm for predicting protein mutant stability changes: application to prion proteins. Protein Eng. 13(12), 849–856 (2000)
12. GriPhyN. Grid Physics Network in ATLAS, http://www.usatlas.bnl.gov/computing/grid/griphyn/
13. Guerois, R., Nielsen, J., Serrano, L.: Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. J. Mol. Biol. 320(2), 369–387 (2002)
14. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 34(web server issue), 729–732 (2006)
15. Khan, S., Vihinen, M.: Performance of protein stability predictors. Hum. Mutat. 31(6), 675–684 (2010)
16. Kinsy, M., Lacroix, Z., Legendre, C., Wlodarczyk, P., Yacoubi, N.: ProtocolDB: Storing Scientific Protocols with a Domain Ontology. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 17–28. Springer, Heidelberg (2007)
17. Kumar, M., Bava, K., Gromiha, M., Prabakaran, P., Kitajima, K., Uedaira, H., Sarai, A.: ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions. Nucleic Acids Res 34, D204–D220 (2006)
18. Kurgan, L., Cios, L., Chen, K.: SCPRED: accurate prediction of protein structural class for sequences of twilight zone similarity with predicting sequences. BMC Bioinformatics 9, 226 (2008)
19. Kwasigroch, J.M., Gilis, D., Dehouck, Y., Rooman, M.: PoPMuSiC, rationally designing point mutations in protein structures. Bioinformatics 18, 1701–1702 (2002)
20. Lacroix, Z., Aziz, M.: Resource descriptions, ontology, and resource discovery. International Journal of Metadata, Semantics and Ontologies 5(3), 194–207 (2010)
21. Lacroix, Z., Legendre, C., Tuzmen, S.: Reasoning on Scientific Workflows. In: Proceedings of the IEEE International Workshop on Scientific Workflows, vol. World Conference on Services - I, pp. 306–313. IEEE Computer Society (2009)
22. Lacroix, Z., Ludaescher, B., Stevens, R.: Integrating Biological Databases. In: Bioinformatics - From Genomes to Therapies, vol. III, pp. 1525–1572. Wiley-VCH Publisher (2007)
23. Letondal, C.: A web interface generator for molecular biology programs in Unix. Bioinformatics 17, 73–82 (2001)

24. Lonquety, M., Lacroix, Z., Papandreou, N., Chomilier, J.: SPROUTS: a database for the evaluation of protein stability upon point mutation. Nucleic Acids Res. 37, 374–379 (2009)
25. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific Workflow Management and the KEPLER System. Concurrency and Computation: Practice and Experience, Special Issue on Scientific Workflows 18(10), 1039–1065 (2005)
26. Maechling, P., Chalupsky, H., Dougherty, M., Deelman, E., Gil, Y., Gullapalli, S., Gupta, V., Kesselman, C., Kim, J., Mehta, G., Mendenhall, B., Russ, T., Singh, G., Spraragen, M., Staples, G., Vahi, K.: Simplifying construction of complex workflows for non-expert users of the southern california earthquake center community modeling environment. ACM SIGMOD Record 34(3), 24–30 (2005)
27. Magyar, C., Gromiha, M., Pujadas, G., Tusnády, G., Simon, I.: SRide: a server for identifying stabilizing residues in proteins. Nucleic Acids Res. 33, W303–W305 (2005)
28. Masso, M., Vaisman, I.: Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. Bioinformatics 24, 2002–2009 (2008)
29. McPhillips, T.M., Bowers, S.: An approach for pipelining nested collections in scientific workflows. ACM SIGMOD Record 34(3), 12–17 (2005)
30. McPhillips, T., Bowers, S., Ludäscher, B.: Collection-Oriented Scientific Workflows for Integrating and Analyzing Biological Data. In: Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS (LNBI), vol. 4075, pp. 248–263. Springer, Heidelberg (2006)
31. Medeiros, C.B., Perez-Alcazar, J., Digiampietri, L., Pastorello, J.G.Z., Santanche, A., Torres, R.S., Madeira, E., Bacarin, E.: WOODSS and the Web: annotating and reusing scientific workflows. ACM SIGMOD Record 34(3), 18–23 (2005)
32. Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T., Goble, C.: Taverna, reloaded. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 471–481. Springer, Heidelberg (2010)
33. Néron, B., Ménager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P., Letondal, C.: Mobyle: a new full web bioinformatics framework. Bioinformatics 22, 3005–3011 (2009)
34. Parthiban, V., Gromiha, M., Schomburg, D.: CUPSAT: prediction of protein stability upon point mutation. Nucleic Acids Res. 34, W239–W242 (2006)
35. Pokala, N., Handel, T.: Energy Functions for Protein Design: Adjustment with Protein–Protein Complex Affinities, Models for the Unfolded State, and Negative Design of Solubility and Specificity. J. Mol. Biol. 347(1), 203–227 (2005)
36. Potapov, V., Cohen, M., Inbar, Y., Schreiber, G.: Accurate structure modelling based on precise description of inter-residue interactions. BMC Bioinformatics 11, 374 (2010)
37. Potapov, V., Cohen, M., Schreiber, G.: Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. Protein Eng. Des. Sel. 22(9), 553–560 (2009)
38. Religa, T.L., Markson, J.S., Mayor, U., Freund, S.M.V., Fersht, A.R.: Solution structure of a protein denatured state and folding intermediate. Nature 437, 1053–1056 (2005)
39. Rohl, C., Strauss, C., Misura, K., Baker, D.: Protein structure prediction using Rosetta. Methods Enzym. 383, 66–93 (2004)
40. Sarachu, M., Colet, M.: wEMBOSS: a web interface for EMBOSS. Bioinformatics 21, 540–541 (2005)

41. Schymkowitz, J., Borg, J., Stricher, F., Nys, R.F., Serrano, L.: The FoldX web server: an online force field. Nucleic Acids Res. 33, W382–W388 (2005)
42. Tokuriki, T.D., Stability, N.: effects of mutations and protein evolvability. Curr. Opin. Struct. Biol. 19(5), 596–604 (2009)
43. Tufféry, P., Lacroix, Z., Ménager, H.: Semantic Map of Services for Structural Bioinformatics. In: Proc. 18th International Conference on Scientific and Statistical Database Management, pp. 217–224. IEEE, Vienna (2006)
44. Wainreb, G., Ashkenazy, H., Bromberg, Y., Starovolsky-Shitrit, A., Haliloglu, T., Ruppin, E., Avraham, K., Rost, B., Ben-Tal, N.: Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. Bioinformatics 27, 3286–3292 (2011)
45. Wainreb, G., Ashkenazy, H., Bromberg, Y., Starovolsky-Shitrit, A., Haliloglu, T., Ruppin, E., Avraham, K., Rost, B., Ben-Tal, N.: MuD: an interactive web server for the prediction of non neutral substitutions using protein structural data. Nucleic Acids Res. 38, W523–W528 (2010)
46. Worth, C., Preissner, R., Blundell, T.: SDM - a server for predicting effects of mutations on protein stability and malfunction. Nucleic Acids Res. 39, W215–W222 (2011)
47. Yang, Y., Zhou, Y.: Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all atom statistical energy functions. Prot. Sci. 17, 1212–1219 (2008)
48. Yin, S., Ding, F., Dokholyan, N.: Eris: an automated estimator of protein stability. Nature Meth. 4, 466–467 (2007)
49. Zhou, H., Zhou, Y.: Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci. 11, 2714–2726 (2002)
50. Zhou, H., Zhou, Y.: Fold recognition by combining sequence profiles derived from evolution and from depth dependent structural alignment of fragments. Proteins 58, 321–328 (2005)

Estimating Viral Haplotypes in a Population Using k-mer Counting

Raunaq Malhotra[1], Shruthi Prabhakara[1], Mary Poss[2], and Raj Acharya[1]

[1] Department of Computer Science and Engineering
Pennsylvania State University, University Park, PA, 16801, USA
[2] Department of Biology, Center for Infectious Disease Dynamics
Pennsylvania State University, University Park, PA, 16801, USA
{rom5161,sap263,acharya}@cse.psu.edu, mposs@bx.psu.edu

Abstract. Viral haplotype estimation in a population is an important problem in virology. Viruses undergo a high number of mutations and recombinations during replication for their survival in host cells and exist as a population of closely related genetic variants. Due to this, estimating the number of haplotypes and their relative frequencies in the population becomes a challenging task. The usage of a sequenced reference genome has its limitations due to the high mutational rates in viruses. We propose a method for estimating viral haplotypes based only on the counts of k-mers present in the viral population without using the reference genome. We compute k-mer pairs that are related to each other by one mutation, and compute a minimal set of viral haplotypes that explain the whole population based on these k-mer pairs. We compare our method to the software ShoRAH (which uses a reference genome) on simulated dataset and obtained comparable results, even without using a reference genome.

Keywords: viral haplotype estimation, structural variants detection, k-mer counting, variant detection, greedy generating set algorithm.

1 Introduction

Viruses only replicate within living cells of a host-organism to form a viral population. The within-host virus population consists of a collection of closely related genetic variants, known as quasi species, wherein the genetic variants occur with different relative frequencies. The genetic variability of these haplotypes is due to the high rate of mutations, resulting in insertions, deletions and substitutions, in the genomes of existing viruses.

Viral population reconstruction involves identification of the genetic variants of the virus present in a viral population. The high genetic diversity of a pathogen population has important consequences in disease progression. It allows the virus to evade host defenses and confounds preventative and therapeutic interventions. The toll of viral evolution on prevention effort is exemplified by the influenza virus; new vaccines must be formulated annually to keep abreast of the seasonally circulating strains of this virus. It is important to reconstruct the different

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 265–276, 2013.

haplotypes and their relative frequencies in a viral population to understand pathogenesis, for drug design and to develop effective public health intervention strategies. Because of their high replication rates, simple genomes, large population sizes, and high mutation and recombination rates, viruses make good models for studying and testing the evolutionary theory.

Next Generation Sequencing (NGS) technologies have opened up an array of possibilities for characterization of genetic diversity in viral populations. NGS technologies generate a large number of genomic sequences (also known as reads) efficiently and economically. Typically, one obtains multiple random copies of the genomic sequences covering all parts of the viral genomes. The high coverage and enormous sequence data output by NGS technologies has the potential to resolve the genetic variation within the virus sample and thereby infer the population dynamics and structure[9].

2 Related Work

A number of methods have been published for viral population reconstruction [1,10,11]. A survey of viral haplotype estimation methods can be found in [2]. Haplotype estimation (viral population reconstruction) can be performed locally along segments of the viral genome or globally across the whole genome. The local haplotype estimation is based on first aligning the reads to a reference genome and then estimating the number of haplotypes. The global estimation of the haplotypes is based on a graph theoretic solution, wherein a set of haplotypes were obtained by calculating a minimal coverage set of paths over a graph of aligned reads [8,18,16]. Probabilistic methods for estimating the haplotypes have been explored in [14,18]. The frequency of individual haplotypes can be computed using an expectation-maximization (EM) algorithm [14,8,18,16].

However, all of the methods rely on the existence of an assembled reference genome. This limits their use to well studied viruses. An imperfect alignment to an inaccurate reference genome due to sequencing errors and high mutational rates in viruses further restricts their usage. In this paper, we propose a method for reconstructing viral haplotypes in a population based on counting the k-mers observed in the viral population without using a reference genome.

Our method is based on the fact that within a population, the viral haplotypes occur in an equilibrium distribution of closely related haplotypes [7]. The viral haplotypes can be changed from one to another by making mutational changes in either of the viral haplotypes. Thus, if the k-mers obtained from a read sampled from one haplotype, is changed by a few mutations (insertion, deletion or substitution) to another k-mer observed in the viral population, then the two k-mers capture a genetic variant of the population. We define these two k-mers as a k-mer pair. This assumes that a k-mer pair maps uniquely in the genomic sequence, and few changes does not leads to the k-mers in a pair mapping to a different location of the genome. This is true if the value of k is large and thus one can determine such mutationally related k-mer pairs. As the number of sampled reads (or k-mers) follows a Poisson distribution [12], we estimate a set

of occurrences which explain all the observed k-mer pairs. We finally estimate a minimal set of haplotypes that explains all the mutationally related k-mer pairs based on a greedy heuristic algorithm proposed in [13]. The method does not depend on the presence of a sequenced reference genome of the viral population, and only requires the counts of individual k-mers present in the viral population. This gives a unique advantage to our method, as it can predict the haplotypes based on intrinsic information present in the viral population.

We evaluate our method over viral populations of varying diversity and population depths and compare our results to that obtained from the software ShoRAH [17]. The number of predicted haplotypes and their frequencies by our method matches closely with those obtained from ShoRAH. ShoRAH provides a large number of false positive viral haplotypes, while our method provides a minimal set that explains all the reads.

The paper is organized as follows: Section 3 describes the methodology for computing the viral haplotypes in the population based on k-mer counting. We define the meaning of mutationally related k-mer pairs and describe an algorithm for inferring their occurrence values based on a mixture of Poisson distributions. Section 4 describes the results obtained from simulated data from HIV samples. We conclude the paper with a summary and discussion of future extensions of this work in Section 5.

3 Methods

Let the viral haplotypes in a population be denoted as the set $\mathbf{VP}$,

$$\mathbf{VP} = \{\mathbf{H_1}, \mathbf{H_2}, ..., \mathbf{H_K}\}, \tag{1}$$

where K is the number of haplotypes in the population. For simplicity, we assume that each haplotype $\mathbf{H_i}$ is of length G, that is,

$$\mathbf{H_i} = \{h_{i1}h_{i2}...h_{iG}\}, \tag{2}$$

where $h_{ij} \in \{A, G, C, T\}$. The haplotype $\mathbf{H_i}$ occurs with relative frequency of α_i in the population, such that

$$\sum_{i=1}^{K} \alpha_i = 1. \tag{3}$$

We denote the set of N reads obtained from the population by $\mathbf{R} = \{R_1, R_2, ..., R_N\}$, where each read is of length L. Typically, the size of the genome G is much larger than the read length L, which depends on the sequencing technology used for obtaining the samples. The reads may contain sequencing errors in the form of substitutions, deletions, and insertions depending on the sequencing technology. Also the read length L can be an average length across all the reads.

Our task is to estimate the number of haplotypes K, their genomic sequences (set $\mathbf{VP}$), and their relative frequencies (α_i) based on the read set $\mathbf{R}$.

We estimate the number of haplotypes and their genomic sequences based on counting k-mers present in the read set $\mathbf{R}$. We use k-mers as they provide a

better resolution as compared to the reads to find mutations amongst the viral haplotypes. The k-mers are classified into three groups based on their relative counts as erroneous, possible variants, and dominant haplotype k-mers. As the reads across the genome follow a Poisson distribution based on the coverage depth, we model the possible variant and dominant haplotype groups as a mixture of Poisson distributions and estimate their means. We also compute the relationship between individual k-mers in each group to find pairs of related k-mers. We estimate a minimal set of viral haplotypes from the estimated means based on a greedy generating set algorithm [6].

Our method is related to an algorithm, previously proposed by us, MutantBin [13], wherein we compute the means of the Poisson curves based on a variable bandwidth mean-shift algorithm. In this paper, we compute the means of the Poisson curves based on the relationship of k-mers to each other and the estimate the Poisson means from the counts of the related k-mer pairs. We implement the greedy generating set algorithm for estimating the minimal set of haplotypes from the means of the related k-mers.

We next describe our method and its assumptions in detail . We describe an algorithm to compute the means of the Poisson mixtures based on the related k-mers, and then describe the algorithm for computing the generating set based on the estimated means.

3.1 Assumptions and Definitions

We assume haplotypes in the viral population are closely related to each other. In other words, a haplotype $\mathbf{H_i}$ can be transformed to haplotype $\mathbf{H_j}$ by changing certain bases, $\{i_1, i_2, ..., i_p\}$ in haplotype $\mathbf{H_i}$. Thus, if we change the bases $\{h_{ii_1}, h_{ii_2}, ..h_{ii_p}\}$ in $\mathbf{H_i}$ to a value from the set $M = \{A, G, C, T, -\}$, we will obtain the haplotype $\mathbf{H_j}$. The number p for any two haplotypes in the viral population is small, but can vary from different populations and to the variants being considered. The $-$ in the set M denotes a gap or removal of a nucleotide from a haplotype $\mathbf{H_i}$ when transforming it into another haplotype $\mathbf{H_j}$. This assumption is valid, as the different viral haplotypes are obtained from high mutational rate during replication of viruses in the population [4].

For example, consider a viral population containing three haplotypes as depicted in Figure 1. The differences in the haplotypes are highlighted by their colors, wherein haplotype A has a "G" at position 5, while haplotypes B and C have a "T" nucleotide. Also haplotype A differs from haplotype B by a "A" at position 14, and a "G" at position 23. Thus, changing these three bases would transform the haplotype A to haplotype B.

A k-mer is a sequence of consecutive k-bases in a read obtained from the read set $\mathbf{R}$. The read set $\mathbf{R}$ contains multiple reads from all parts of the haplotype set $\mathbf{V}$, therefore, the k-mers obtained from the reads will also span all parts of the haplotype set. A k-mer corresponds to a unique region in a haplotype $\mathbf{H_i}$ as long as the value of k is sufficiently large and there are no repeat regions in the genome. Indeed, one does not observe repeats in viruses, and choosing a

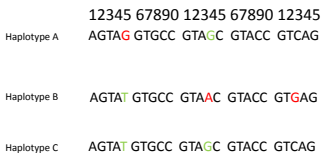


Fig. 1. An example viral population containing 3 different haplotypes A,B and C, where the differences in them are colored in red compared to the others

large value of k (> 20 bp) ensures unique mapping of k-mers to a region of the haplotype.

The number of times a k-mer is observed corresponds to the coverage of the viral haplotype constituting it. This is because the number of times a position i in the genome $\mathbf{H_i}$ is sampled follows a Poisson distribution with mean value equal to the coverage of the genome [12]. Thus, a k-mer sampled from a region in the haplotypes $\mathbf{H_i}$ which is common to all other haplotypes $\mathbf{H_j}$ in the viral population set $\mathbf{V}$ will be sampled from a Poisson distribution with mean equal to the sum of abundances (or coverages) of each population.

Interestingly, if the k-mer corresponding to the unique region in haplotype $\mathbf{H_p}$ is transformed to the k-mer that is common amongst all the other haplotypes $\{\mathbf{H_j} : j \neq i\}$, then the sum of the abundances of these two k-mers will also be sampled from a Poisson distribution of mean equal to the sum of abundances of each population. However, a k-mer corresponding to a unique region of a particular haplotype, say $\mathbf{H_p}$, would occur from a Poisson distribution with mean value of abundance of that haplotype. Also, if a k-mer contains an error from sequencing or contamination of the sample, the k-mer will be observed a few number of times in the viral population.

3.2 Computation of Related k-mer Pairs and Estimation of Poisson Parameters

We can estimate a minimum number of viral haplotypes that are required to explain all the k-mers observed in the viral population. The histogram of observed k-mers can be plotted to visualize the mixture of Poisson distributions observed in the population. An example of such a histogram is shown in Figure 2, wherein the histogram is obtained by counting 21-mers from a simulated viral population containing 3 different haplotypes.

The set of k-mers observed in a viral population can also be classified into three distinct groups based on their counts of occurrence. The k-mers with very low counts correspond to sequencing errors, and constitute the first group (Group A). We assume that Group A does not contain k-mers corresponding to even the lowest abundant viral haplotype in the population. These k-mers occur on the left side of the k-mer histogram. We consider all k-mers that occur below a certain threshold of occurrence as errors. At the other extreme, the k-mers

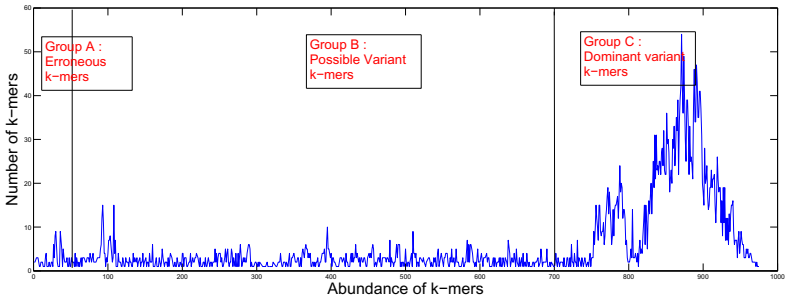


Fig. 2. Abundance plot of 21-mers obtained from a simulated viral population containing 3 haplotypes of hiv-1 glycoprotein (env) gene

that are observed in all haplotypes in the population, will have a high coverage, and occur on the right end of the histogram. These constitute the second group of k-mers (Group C), and provide information about the coverage had there been only a single viral haplotype in the population. The k-mers observed with intermediate counts constitute the last group (Group B), and correspond to regions of mutations amongst the viral haplotypes. The boundary between Group B and C can be determined empirically, based on the fact that k-mers in group B can be transformed into k-mers of Group C.

A k-mer present in one group can be transformed into a k-mer of the other group based on one or two mutations. This is easy to see for k-mers belonging in Group A. A change in one nucleotide of a k-mer in Group A might match it to a variant region (Group B) in the viral haplotype or to the common region amongst all the haplotypes (Group C). Similarly, the k-mers in group B can be transformed to k-mers in group C by mutational changes.

We model the distribution of k-mers as a mixture of Poisson distributions. An important first step for that is inferring the number of Poisson distributions that represent all the error-free k-mers in the population. One can infer the number of Poisson distributions present in groups B and C of k-mers by observing the occurrences of pairs of related k-mers. These pairs of k-mers capture the local viral haplotype variants present in the population.

The algorithm for estimating the Poisson distributions parameters is described in Algorithm 1. The basic idea is that the number of Poisson distributions present is bounded by the pairs of related k-mers observed in the population, and that all occurrence values within two standard deviations of the mean of a Poisson distribution belong to that particular distribution. This is because the probability of a value to lie within two standard deviations of the mean is close to one for large range of mean values.

The computation of the k-mer counts from the reads is linear in the number of reads, while finding the pairs of related k-mers in step 2 has time complexity $O(|R|k)$. The computation of Poisson means is linear in the number of k-mers, making the overall complexity of the algorithm to be $O(|R|k)$.

Algorithm 1. Algorithm for inferring number of Poisson distribution mixtures in the k-mer counting

Input: $\mathbf{R} = \{R_i\}_{i=1}^{N}$, value k to be used for k-mer counting
Output: A number of Poisson distribution means representing all pairs of related k-mers

1. Compute counts of all k-mers present in the read set $\mathbf{R}$. Denote the set of k-mers as $\mathbf{V}$, and count of a k-mer v as $C(v)$.
2. For each $v \in \mathbf{V}$
   (a) Transform v by single nucleotide changes to another k-mer, $u \in \mathbf{V}$, such that $C(u) > C(v)$. Associate all such k-mers u to v. Denote the set as S_v.
   (b) Store the counts of the k-mers in set S_v to a collective set $\mathbf{B}$.
3. Estimation of means of Poisson distributions from the set $\mathbf{B}$

```
Sort the values in set B
P= (); # Set of Poisson Means
for b in set B
        b_found = 0
        foreach p in P
                if( abs(b-p) < 2 sqrt(p) )
                        b_found = 1
       if(b_found ==0 )
                P = [P;b]
return P
```

3.3 Greedy Algorithm for Minimal Haplotype set Estimation

Once we obtain the set of means corresponding to the various local haplotype variants we can infer the haplotypes globally based on a greedy heuristic as proposed in [13]. The greedy approach estimates the minimal number of haplotypes explaining the set of Poisson means by formulating the problem as a minimal generating set problem with no repeats. This generating set problem was proven to be NP complete [6].

The generating set algorithm is described in Algorithm 2. The algorithm takes as input a set of numbers corresponding to the means of the observed Poisson distributions and outputs a minimal set of numbers corresponding to the frequencies of the viral haplotypes. The input numbers can be explained by sums of combinations of output numbers. Here the set P denotes the set of means obtained from Algorithm 1, while the set X denotes the set of output frequencies for the viral haplotypes. The algorithm starts with an empty set for X. Next it traverses through the set P in increasing order, and adds numbers to the set X only if the current number in P cannot be explained by sums of numbers present in set X. The aim is that the number added in X should explain as many possible elements in P as possible. The difference set D makes sure that we do not remove the most common difference from the set P.

Algorithm 2. Algorithm for computing the generating set (minimal set of explaining viral haplotypes)

Input: $\mathbf{P} = \{p_1, p_2, ..., p_n\}$ set of means from Algorithm 1
Output: $X = \{x_1, x_2, .., x_K\}$ set of abundances of the K haplotypes in the population

1. Initialize set $X = \{\phi\}$, $T \leftarrow \mathbf{P}$
2. Create difference set $D = \{$All differences in set P$\}$
3. While the set $T \neq \phi$
   (a) Add the minimum value of set T to X and remove it from T
   (b) For all subsets of X, compute the sum of values in the subset and see if it belongs in the set T. If it does, and the sum is not a mode of the difference set D, then remove it from set T.
4. Return set X

4 Results

We evaluate our method on a number of simulated datasets of varying levels of diversity, both in the number of haplotypes present and their relative similarity. The similarity between two viral haplotypes is defined based on the pairwise comparison of the haplotypes [13]. The simulated datasets were generated from HIV samples. The viral population consisted of haplotypes of 2000 bp fragment of HIV-1 genome from the 5' end, which were obtained by using the population sampler toolkit in sequencing simulation software Metasim [15].

We simulated four datasets from the HIV-1 genome with varying degrees of diversities. The diversity of a sample is defined as percentage of bases that are mutations amongst the population. Three of these four datasets contain two viral haplotypes with different relative frequencies, while one of them contains three viral haplotypes with relative frequencies of 1:3:5. The details of the simulated datasets are listed in Table 1. Datasets 1-3 contain populations of lengths (1000, 2000 and 4000 bps) and diversities varying between 0.2% to 10% in steps of 0.2% (overall 150 populations each). Dataset 4 contains populations of length 1000bp and diversity varying between 0.2% to 5% in steps of 0.2% (25 populations). The first three datasets were generated to evaluate the performance of our algorithm in reproducing the relative frequencies of the viral haplotypes when the dominant virus is more prevalent, while the fourth evaluates the ability of our algorithm to resolve more than two viral haplotypes in a population.

We simulated 454-Roche sequencing technology reads for each of the viral populations using the simulation software Metasim [15]. All the simulation settings except the insert size were kept at default values. We simulated 10,000 to 50,000 reads for each of the four datasets. We conduct experiments with different values of k (13,15,17,21,23,25). We set the value of k for computing k-mer counts to 21 as it provides the best results based on F-score values. The value of k should be large enough so that every k-mer maps uniquely to a reference genome of the virus. We next compute the pairs of mutationally related k-mers. Such k-mer pairs are computed by finding all one-two mutation versions of a k-mer and

Table 1. Statistics of the simulated datasets used for evaluation of our method. The diversity is computed as an average of all pairs pairwise distances of the haplotypes in the population [13].

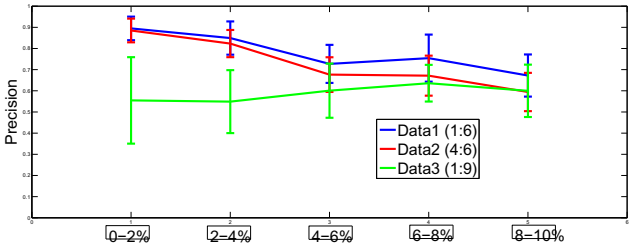
Data source	Diversity in the sample	Number of haplotypes	Relative frequencies
1. HIV-1 2000bp	0.2% to 10%	2	1:6
2. HIV-1 2000bp	0.2% to 10%	2	4:6
3. HIV-1 2000bp	0.2% to 10%	2	1:9
4. HIV-1 2000bp	0.2% to 5%	3	1:3:5

associating it to the k-mer which has the highest occurrence in the population. Next we estimate the means of mixture of Poisson distributions based on Algorithm 1. We use a threshold occurrence of 5, below which every k-mer is considered an error. This value is chosen to be the first minimum in the abundance plot. We estimate the number of viral haplotypes in the population and their frequencies using the greedy algorithm proposed in [13].

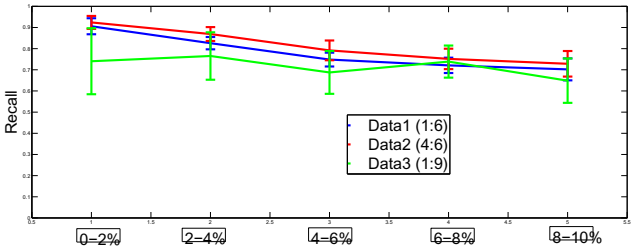
We assign k-mers to different viral haplotypes based on their pairings to the related k-mers. Two k-mers in a pair end up in different viral haplotypes. The k-mers which are present in all the viral haplotypes are assigned to each of the viral haplotypes. We compute the precision and recall values for the assignment of k-mers into different haplotypes. The precision is defined as the ratio of the number of correctly assigned k-mers to the total number of k-mers assigned to a viral haplotype. Recall is defined as the ratio of the number of correctly assigned k-mers to the number of true k-mers present in a viral haplotype.

Figure 3 summarizes the precision and recall values for viral populations for datasets 1-3. We observe high recall and precision values for populations with small diversity while it is difficult to decipher populations with higher diversity using this particular method. These results are as expected as the diversity in population increases, the number of mutations per k-mer required for computing the k-mer pairs also increases. Thus, we observe that for the particular value of k (21) we are able to resolve low diversity populations quite accurately. We observe that the precision and recall values for dataset 3 have a high variance and are in general low. This is because the minimum number of haplotypes predicted by our algorithm for this dataset was 3 as compared to 2 present in the data. This led to consistent low values for precision and recall for dataset 3.

We also compare the relative frequencies of the haplotypes predicted by our method to those predicted by the software ShoRAH. Table 2 shows the comparison results for datasets 1-3. For datasets 1 and 2, our method predicts a minimum two haplotypes for populations of all diversities, and predicts 3 haplotypes for 45 out of 150 datasets. Thus our generating set algorithm provides correct solution in 70% of the datasets. ShoRAH on the other hand predicts more than two haplotypes for all the three datasets. There are a large number of false positives predicted by ShoRAH. For comparison purposes, we report the number of cases (numbers inside brackets) in which the top two predicted haplotypes from ShoRAH explain more than 95 % of the reads. Our method outperfoms ShoRAH in two out of three datasets.



(a) Precision values for Data1-3



(b) Recall values for Data1-3

Fig. 3. Precision and recall values for datasets 1-3. Each data point contains the precision and recall values computed over 30 populations, varying within the percentage indicated on the x-axis.

Dataset 4 contains 3 viral haplotypes with varying degrees of diversity. Our method reproduces the relative frequencies of the haplotypes accurately in 22 out of 25 runs.

Table 2. Comparison of relative frequencies of populations as predicted by K-mer pairing and ShoRAH. The number in brackets indicate the number of populations for which the predictions contain more than 95% of the reads.

Data sets	K-mer pairing	ShoRAH
Data 1 (1:6)	1:5.52 (150/150)	1:5.77 (123/150)
Data 2 (4:6)	4:7.86 (150/150)	4:6.04 (123/150)
Data 3 (1:9)	1:7.01 (105/150)	1:9 (118/150)

5 Conclusion and Future Work

We have proposed a method for predicting the viral haplotypes in a population without using the reference genome. We use the information from the counts of k-mers observed in the population for inferring the viral haplotypes. Our method improves haplotype identification compared to the software ShoRAH even without using the reference genome. It provides a minimal set of haplotypes

that explains all the reads in the population. We have not performed assembly of the k-mers that are clustered together in this paper. The major challenge in assembly of viral haplotypes is resolving the k-mers into individual haplotypes. As our method provides clustering of k-mers into individual haplotypes, one can obtain the viral haplotype genome by performing *de-novo* assembly of the k-mers, and thus the reads.

The next step will be to apply our method on real datasets, which are more complex and might contain several haplotypes. The presence of sequencing bias in the NGS technologies may affect our method on real datasets. It is possible that common k-mers from all haplotypes cannot be modeled by a single Poisson distribution. There are methods available for correcting the sequencing bias, which can be employed [3]. Binning and smoothening techniques have been used for compensating the GC content bias in the sequenced reads [5]. Moreover, the sequencing technologies are working on reducing the sequencing bias.

That being said, our algorithm is a work in progress. As our method tries to find a minimal set of haplotypes that explain the reads, it is challenging to resolve haplotypes which occur with same relative abundances. The k-mer pairs would get associated with same Poisson peaks, making it difficult to resolve them. We can use reads information to guide our haplotype reconstruction by eliminating haplotypes which are not seen amongst the reads. We have not considered recombinations amongst viral populations in this work. For future work, we would like to incorporate the assumptions of recombination amongst the viral population for predicting the viral haplotypes. Nevertheless, we believe that estimating the viral haplotypes on the basis of counts of k-mers is a direction that should be pursued due to their high mutational and recombination rates.

References

1. Astrovskaya, I., Tork, B., Mangul, S., Westbrooks, K., Măndoiu, I., Balfe, P., Zelikovsky, A.: Inferring viral quasispecies spectra from 454 pyrosequencing reads. BMC Bioinformatics 12(6) (2011)
2. Beerenwinkel, N., Gunthard, H.F., Roth, V., Metzner, K.J.: Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. Frontiers in Microbiology 329(3) (2012)
3. Benjamini, Y., Speed, T.P.: Summarizing and correcting the gc content bias in high-throughput sequencing. Nucleic Acids Research 40(10), e72 (2012)
4. Boerlijst, M.C., Bonhoeffer, S., Nowak, M.A.: Viral quasi-species and recombination. Proceedings of the Royal Society of London. Series B: Biological Sciences 263(1376), 1577–1584 (1996)
5. Boeva, V., Zinovyev, A., Bleakley, K., Vert, J.-P., Janoueix-Lerosey, I., Delattre, O., Barillot, E.: Control-free calling of copy number alterations in deep-sequencing data using gc-content normalization. Bioinformatics 27(2), 268–269 (2011)
6. Collins, M.J., Kempe, D., Saia, J., Young, M.: Nonnegative integral subset representations of integer sets. Inf. Process. Lett. 101, 129–133 (2007)
7. Eigen, M., McCaskill, J., Schuster, P.: The molecular quasi-species. Adv. Chem. Phys. 75, 149–263 (1989)

8. Eriksson, N., Pachter, L., Mitsuya, Y., Rhee, S.-Y., Wang, C., Gharizadeh, B., Ronaghi, M., Shafer, R.W., Beerenwinkel, N.: Viral population estimation using pyrosequencing. PLoS Comput. Biol. 4(5), e1000074 (2008)
9. Hoffmann, C., Minkah, N., Leipzig, J., Wang, G., Arens, M.Q., Tebas, P., Bushman, F.D.: DNA bar coding and pyrosequencing to identify rare HIV drug resistance mutations. Nucleic Acids Research 35, 91 (2007)
10. Jojic, V., Hertz, T., Jojic, N.: Population sequencing using short reads: HIV as a case study. In: Proc. Pac. Symp. Biocomput., pp. 114–125 (2008)
11. Macalalad, A.R., Zody, M.C., Charlebois, P., Lennon, N.J., Newman, R.M., Malboeuf, C.M., Ryan, E.M., Boutwell, C.L., Power, K.A., Brackney, D.E., Pesko, K.N., Levin, J.Z., Ebel, G.D., Allen, T.M., Birren, B.W., Henn, M.R.: Highly sensitive and specific detection of rare variants in mixed viral populations from massively parallel sequence data. PLoS Comput. Biol. 8(3), e1002417 (2012)
12. Port, E., Sun, F., Martin, D., Waterman, M.S.: Genomic mapping by end characterized random clones: A mathematical analysis. Genomics 26, 84–100 (1995)
13. Prabhakara, S., Malhotra, R., Poss, M., Acharya, R.: Mutant Bin: Unsupervised Haplotype Estimation of Viral Population Diversity Without Reference Genome. Journal of Computational Biology (in press)
14. Prosperi, M., Prosperi, L., Bruselles, A., Abbate, I., Rozera, G., Vincenti, D., Solmone, M., Capobianchi, M., Ulivi, G.: Combinatorial analysis and algorithms for quasispecies reconstruction using next-generation sequencing. BMC Bioinformatics 12, 5 (2011)
15. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: Metasim: A sequencing simulator for genomics and metagenomics. PLoS One 3, 3373 (2008)
16. Westbrooks, K., Astrovskaya, I., Campo, D., Khudyakov, Y., Berman, P., Zelikovsky, A.: HCV quasispecies assembly using network flows. In: Măndoiu, I., Wang, S.-L., Zelikovsky, A. (eds.) ISBRA 2008. LNCS (LNBI), vol. 4983, pp. 159–170. Springer, Heidelberg (2008)
17. Zagordi, O., Bhattacharya, A., Eriksson, N., Beerenwinkel, N.: ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. BMC Bioinformatics 12(1), 119 (2011)
18. Zagordi, O., Geyrhofer, L., Roth, V., Beerenwinkel, N.: Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction. Journal of Computational Biology 17(3), 417–428 (2010)

Fast Computation of Entropic Profiles for the Detection of Conservation in Genomes

Matteo Comin and Morris Antonello

Department of Information Engineering, University of Padova,
Via Gradenigo 6/A, Padova, Italy
comin@dei.unipd.it

Abstract. The information theory has been used for quite some time in the area of computational biology. In this paper we discuss and improve the function Entropic Profile, introduced by Vinga and Almeida in [23]. The Entropic Profiler is a function of the genomic location that captures the importance of that region with respect to the whole genome. We provide a linear time linear space algorithm called Fast Entropic Profile, as opposed to the original quadratic implementation. Moreover we propose an alternative normalization that can be also efficiently implemented. We show that Fast EP is suitable for large genomes and for the discovery of motifs with unbounded length.

Keywords: pattern discovery, information theory, computational biology.

1 Introduction

The concept of information theory was originally introduced by Claude E. Shannon as a tool to systematically analyze data flow in general communication systems [20]. The theory has been extended and subsequently applied to many fields including DNA sequence analysis [24]. Methods of Information theory focusing on DNA sequence compression have found differences between coding and noncoding sequences [17] and they have been applied also for classification [3,4]. In [12] the authors applied the mutual information to discover SNPs that are significantly associated with diseases. Also compression based classification relying on mutual information can be successfully applied to phylogeny [2]. Moreover the identification of splicing mutations can benefit from the use of Information Theory[18]. In [11] sequence motifs are modeled based on the maximum entropy principle. Such models can be utilized to discriminate between signals and decoys. In [5] an entropic segmentation method is discussed to detect borders between coding and noncoding DNA. These are just a few examples of the computational biology applications inspired by information theory.

In this paper we discuss and improve the function Entropic Profile, introduced by Vinga and Almeida in [23]. The concept of Entropic Profiler was introduce to analyze DNA sequences. The Entropic Profiler is a function of the genomic location that captures the importance of that region with respect to the whole genome. This score is based on the Shannon entropies of the words distribution. This method proved useful for the identification of conserved genomic regions.

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 277–288, 2013.

Other types of sequence profile have also been previously explored like Sequence Logos [19], that provide the information content per position. This method, however, requires the alignment of a set of sequences and thus it is not suited for a single sequence. Moreover this approach does not comply to the alignment-free paradigm like [8].

One of the most important requirements is the development of efficient methods for the analysis of whole genomes that can scale gracefully with the size of input. In this paper we study the use of Suffix Tree for the computation of the Entropic Profiler. We show that the same function can be evaluated in linear time and space as opposed to the quadratic implementation of EP [23]. This will allow the use of longer genomes and the discovery of motifs with unbounded length, removing the limitations of the current implementation. Moreover we propose an alternative normalization that can be also efficiently implemented within the Suffix Tree structure. The resulting implementation will be named Fast Entropic Profile (FastEP). We show that FastEP proved useful for the detection of conserved signals.

1.1 Entropic Profiler

Although DNA is a flexible three-dimensional molecule interacting in a dynamic environment, its digital information can be represented by a one dimensional character string of G's, A's, T's and C's. Following this standard assumption, two of its most striking features are the extent to which repeated L-tuples occur and the variety of repeated structures it contains. These topics have been discussed extensively and various mechanisms try to explain the functional and evolutionary role of repeats. The degree of predictability and randomness of a substring is described by its entropy [23]. Entropic Profiles (EP) are plots estimated by this local entropy formulation, defined for each position/symbol, from the complete sequence of DNA. The original definition is based on the distribution of words that end at a particular location i. Let s be the input genome of length $|s| = n$, we define $s[i, i+k-1]$ as the word of length k that starts at position i. Let $c[i, i+k-1]$ be the number of time the word $s[i, i+k-1]$ appears in the genome s. The function local entropy for position i is defined as:

$$g_{L,\phi}(i) = \frac{1 + 1/n \sum_{k=1}^{L} 4^k \phi^k c[i-k+1, i]}{\sum_{k=0}^{L} \phi^k} \tag{1}$$

where ϕ is a normalization parameter. This function can be interpreted as a linear combination of suffix counts up to a given length L, with different weights. It computes, for each location of the sequence, the information about the abundance of the corresponding L-tuple suffix inside the entire sequence. For ease of explanation we redefine the above formula to evaluate the statistic of words starting at position i, instead of ending at position i.

$$f_{L,\phi}(i) = \frac{1 + 1/n \sum_{k=1}^{L} 4^k \phi^k c[i, i+k-1]}{\sum_{k=0}^{L} \phi^k} \tag{2}$$

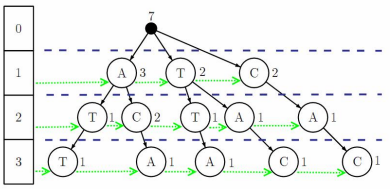


Fig. 1. Truncated suffix tree, L=3, and side links of the word ATTACAC

Note that the function $g_{L,\phi}(i)$ is equivalent to compute $f_{L,\phi}(n-i)$ for the reverse of s. This function is then normalized to allow the comparison of different parameter combinations. EP values are normalized as a z-score: $EP_{L,\phi}(i) = \frac{f_{L,\phi}(i) - m_{L,\phi}}{s_{L,\phi}}$, where the mean is $m_{L,\phi} = \frac{1}{n}\sum_{i=1}^{n} f_{L,\phi}(i)$ and the standard deviation $s_{L,\phi} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(f_{L,\phi}(i) - m_{L,\phi}\right)^2}$.

We will discuss an alternative normalization in section 3. The original implementation of the entropic profiler is based on a truncated suffix trie, see Figure 1. A standard trie, storing the collection of n suffixes of the entire DNA sequence, has the following properties:

- the number of nodes is $O(n^2)$.
- the height is equal to the length of the longest string, that is the length of the whole sequence, n.
- word matching for a pattern of length L takes $O(L)$ time.
- constructing the entire trie takes $O(n^2)$ time.

The counters at each node represent the number of occurrences of the corresponding word. This allows the main EP function to be worked out by simply word matching. All nodes at the same depth are connected by side links in order to speed up the normalization, otherwise the computation of $m_{L,\phi}$ and $s_{L,\phi}$ would involve the repeated calculation of the main EP function for all positions.

There are two problems with this implementation. The first issue is that it is space inefficient. Specifically, there may be a lot of nodes that have only one child, and the existence of such nodes is a waste. The second problem is that the Entropic Profiler can be computed only for small L. In fact in [23] the function EP can explored only for motif shorter than 15 bases, and thus the trie is truncated at depth 15. These observations have prompted the idea to consider instead of a trie its compressed version also known as Suffix Tree.

1.2 Preliminaries on Suffix Trees

The Suffix Tree is one of the most studied data structures and it is fundamental for string processing. It stores a string in such a way that enables the implementation of efficient searches. Traditionally the suffix tree has been used in very different fields, spanning from data compression [26,3] to clustering [10] and classification [9,8]. The use of suffix tree has become very popular in the field of bioinformatics allowing a number of string operations, like detection of

repeats [14], local alignment [16], the discovery of regulatory elements [6,7] and extensible patterns [1]. The optimal construction of suffix tree has already been addressed by [22,15], that provided algorithms in linear time and space. Figure 2 shows an example of suffix tree for the string $s = TCGGCGGCAAC$. We can observe that each suffix of the string s is present in the tree as a labeled path from the root to a leaf.

2 Fast Entropic Profiler

This section we describe how the entropic profiler can be efficiently computed using the suffix tree. Let assume that we have already computed the suffix tree of the input string s using the algorithm of Ukkonen [22]. We extend this structure so that every node v contains a variable $count(v)$ that stores the number of times that the word represented by v appears in s. With a simple $O(n)$ traversal of the tree we can compute the variable $count(v)$ of each internal node v, where $count(l) = 1$ if l is a leaf.

The goal is to find an efficient way to compute the main EP function 2 for every possible substring and parameter combination. If the substring taken into consideration is encoded by the suffix tree, there are two main cases: it may be spelled out by the concatenation of the edge-labels on the path from the root to a node or not. In the latter case the substring ends between two nodes.

The function $f_{L,\phi}(i)$ for each sequence belonging to the former case can be preprocessed and stored in a variable $entropy(v)$, for each node v. Now assume that the node v represents the string $s[i, i + L - 1]$ then the variable $entropy(v)$ will contain $\sum_{k=1}^{L} 4^k \phi^k c[i, i + k - 1]$, the main sum of $f_{L,\phi}(i)$. Once $entropy(v)$ is available we can calculate $f_{L,\phi}(i)$ in constant time. The following preprocessing is a preorder traversal of the tree that computes the value of $entropy(v)$ for all nodes. Let assume that $par(v)$ is the parent node of v, and that $h(v)$ is the length of the string spelled out by the concatenation of the node-labels on the path from the root to that node. In other words $h(v)$ is the length of the string represented by the node v.

Preprocess(T,v)
A suffix tree T and a node v are given.
begin [visit]
if v is the root **then**
$entropy(v) = 0$
else
$entropy(v) = entropy(par(v)) + count(v) \sum_{k=h(par(v))+1}^{h(v)} [4^k \phi^k]$
end if
for all child w of v **do**
begin [recursive traversal]
Preprocess(T,w);
end for

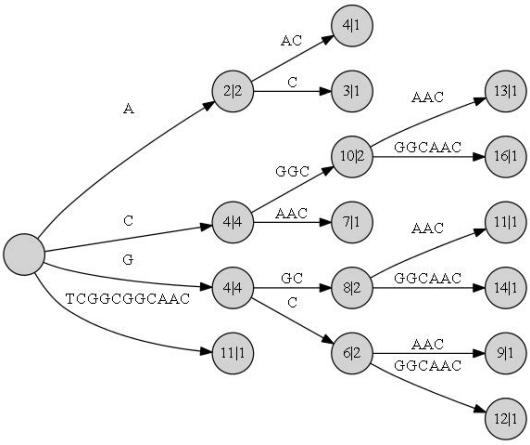


Fig. 2. Suffix tree of the string TCGGCGGCAAC. Every copy of the terminal symbol \$ is removed from the edge labels. The nodes are labeled with the corresponding values of *entropy*|*count*, where for simplicity $4\phi = 1$.

Let's consider the string $TCGGCGGCAAC$ and the suffix tree in Figure 2. The main sum for the function $f_{4,\phi}(2)$ is $\sum_{k=1}^{4} 4^k \phi^k c[2, 2+k-1]$. For ease of explanation we write $c[s[i,j]]$ instead of $c[i,j]$. This sum can be expanded in: $4\phi c[C] + (4\phi)^2 c[CG] + (4\phi)^3 c[CGG] + (4\phi)^4 c[CGGC]$. Now the information contained in the suffix tree allows us to simplify this sum. We can note that every time we see CG it is always followed by a GC, thus $c(CG) = c(CGG) = c(CGGC)$, that is also $count(v)$, where v represent the word $CGGC$. Finally if we consider that $entropy(C) = 4\phi c[C]$ that is also the node $par(v)$. Thus the previous sum can be simplified in : $4\phi c[C] + ((4\phi)^2 + (4\phi)^3 + (4\phi)^4)c[CGGC] = entropy(par(CGGC)) + count(CGGC) \sum_{2}^{4} [4^k \phi^k]$. This is equivalent to the formula used in the preprocessing, where part of the summation is simplified thanks to the suffix tree. Using the properties of the geometric series we can observe that $\sum_{k=h(par(v))+1}^{h(v)} [4^k \phi^k]$ is equivalent to $[(4\phi)^{h(par(v))+1} - (4\phi)^{h(v)+1}]/[1-4\phi]$. Thus each visit takes time $O(1)$, and the total time spent in this preprocessing is $O(n)$, linear the number of nodes.

After this preprocessing, the EP function can be retrieved efficiently for all words represented by some node in the tree T. The following algorithm computes the EP function of any word $s[i, i+L-1]$ of length L using as input the suffix tree T.

FastEP (Input: T, i, L, ϕ; Output:$f_{L,\phi}(i)$)
Search the input word $s[i, i+L-1]$ in the suffix tree T.
if it is represented by the node v **then**
the algorithm **returns** the preprocessed value of the variable $entropy(v)$ of the internal node v.
end if

if the search ends within an edge, between the two nodes u and v **then**
the algorithm **returns** the preprocesses value of $entropy(u)$
plus the correction factor $count(v) \sum_{k=h(u)+1}^{L} 4^k \phi^k$.
end if

In summary if the query word is represented in the suffix tree by a node v it is enough to return $entropy(v)$, otherwise we need to add a correction factor that is proportional to the number of times the word as a whole appears, and thus using $count(v)$. Again from the output of this procedure we can compute in constant time the Entropic Profile function (formula 2). Thus FastEP after a linear time linear space preprocessing can evaluate a certain position or equivalently a specific pattern in constant time. The original implementation requires $O(n^2)$ time and space to answer the same query.

3 Fast Entropic Profiler Normalization

The aim of this section is to provide an alternative normalization of EP such that, in order to be computed, it does not require to process all positions of s and for all L. Algebraic considerations [23] allow the mean $m_{L,\phi}$ to be rewritten as:

$$m_{L,\phi} = \frac{(\phi - 1)(m^2 + \sum_{i=1}^{L} C^2[k])}{m^2(\phi^{L+1} - 1)} \tag{3}$$

where $C^2[k]$ stands for the sum of the squared counts of all distinct words of size k in the whole sequence. Similarly, the standard deviation $s_{L,\phi}$ becomes:

$$s_{L,\phi} = \sqrt{\frac{1}{m-1}\left(\frac{S[L]}{\left(\frac{\phi^{L+1}-1}{\phi-1}\right)^2} - m_{L,\phi}^2 \cdot m\right)} \tag{4}$$

where the recursive function $S[L]$, depending on the number of distinct word of length L, is fairly intricate. Even if L-tuples are less than the length of the whole sequence n, this kind of normalization takes still $O(n^3)$ time and $O(n^2)$ space.

There are several alternatives to the above normalization. In this paper we propose to define FastEP, $FastEP_{L,\phi}(i)$ as :

$$FastEP_{L,\phi}(i) = \frac{f_{L,\phi}(i)}{\mathbf{max}_{0 \le j < n}[f_{L,\phi}(j)]} \tag{5}$$

where the function $\mathbf{max}_{0 \le j < n}[f_{L,\phi}(j)]$ returns the maximum value of $f_{L,\phi}$ over all words of size L. Similarly to the original normalization this formulation allows to compare the entropic profile scores for words of different length. In fact FastEP assumes values in the range $[0, 1]$.

3.1 Finding the Maximum Entropy $f_{L,\phi}$ for all L Using a Branch and Bound Approach

In the following we discuss a branch and bound strategy to efficiently recover the values of $\mathbf{max}_{0\leq j<n}[f_{L,\phi}(j)]$ for all L, or simply $\mathbf{max}_L$. Instead of naively comparing each word of length L, the search for the maximum FastEP can be restricted to some regions of the tree. Again for ease of explanation we will consider only the sum $\sum_{k=1}^{L} 4^k\phi^k c[i, i+k-1]$, as the main $f_{L,\phi}(j)$ can be trivially derived.

If $L > 1$, two definitions are needed to define which regions of the tree must be taken into consideration and which can be pruned:

Definition 1. *The minimum potential maximum mpm_L defines a lower bound to the maximum $f_{L,\phi}(j)$ for all L:*

$$mpm_L = \boldsymbol{max}_{L-1} + 4^L\phi^L$$

Definition 2. *The maximum potential maximum $MPM_L(v)$, where $L > 1$ and v is a node such that $h(v) < L$, is defined as:*

$$MPM_L(v) = entropy(v) + [count(v) - 1] * \sum_{k=h(v)+1}^{L} 4^k\phi^k$$

The maximum potential maximums, MPM bounds, are progressively computed and they allow to prune the search space for the maximum EP. The maximum potential maximum $MPM_L(v)$ is associated to any node v. At each step they define an upper bound to the maximum FastEP obtainable for a path starting from the root and passing trough the node v. In fact, if a $MPM_L(v)$ is less than mpm_L that region can be discarded and not considered. Otherwise if $MPM_L(v)$ is greater than mpm_L we extend this path to the child of v as long as these nodes have height not greater than L.

The following numerical example, which computes the values of $\mathbf{max}_L$ for L from 1 to 2, clarifies these concepts. Let's consider the example of Figure 2 where for simplicity we use $4\phi = 1$. For $L = 1$ it is enough to consider the most frequent character, that is G or C, that produces $\mathbf{max}_1 = entropy(C) = 4$. If L=2 it must be $\mathbf{max}_2 \geq \mathbf{max}_1 + 1 = 5$, where the second term is the minimum potential maximum $mpm_2 = 4 + 1 = 5$. Now for $L = 2$ we have that:

A: $MPM_2(A) = 2 + 1 = 3 < mpm_2 = 5 \rightarrow$ NOT acceptable path;
C: $MPM_2(C) = 4 + 3 = 7 > mpm_2 = 5 \rightarrow$ acceptable path;
G: $MPM_2(G) = 4 + 3 = 7 > mpm_2 = 5 \rightarrow$ acceptable path;
T: $MPM_2(T) = 1 + 1 = 2 < mpm_2 = 5 \rightarrow$ NOT acceptable path;

Two nodes are left out because a priori the maximum for $L = 2$ cannot be found traversing those nodes of the tree. Thus, after following every acceptable path, the value $\mathbf{max}_2$ is worked out by simply comparing:

CA: $entropy(CC) = 4 + 1 = 5$
CG: $entropy(CG) = 4 + 2 = 6$
GC,GG: $entropy(GC) = entropy(GG) = 4 + 2 = 6 \rightarrow \mathbf{max}_2 = 6$

Note that at this step no more nodes are traversed, but since $h(v) < L$ we just take the path with the maximum value of *counts*. In summary we can observe that to obtain $\mathbf{max}_L$ it requires $\mathbf{max}_{L-1}$, thus overall $\mathbf{max}_L$ can be computed in L steps. If $L = n$ in the worse case we can traverse the entire suffix tree, that is $O(n)$ nodes. Thus overall the n values of $\mathbf{max}_L$ can be computed in $O(n^2)$ time and $O(n)$ space. There are some tricks that one can use in the implementation to speedup further this process. We can note that if a node is part of an acceptable path while calculating $\mathbf{max}_L$ it will be also traversed for $\mathbf{max}_{L+1}$. Thus we don't need to traverse that part of the tree from the root, but we can just start from the latest nodes visited for $\mathbf{max}_L$. Another observation is that the value of mpm_L should be reset if the previous maximum ends in a leaf. For comparison with the original approach, based on truncated tries, the normalization process can take $O(n^3)$ time and $O(n^2)$ space, whereas our branch and bound strategy requires $O(n^2)$ times and linear space.

3.2 Expected and Real Efficiency

The expected fraction of nodes in the tree that are pruned can be computed as the following probability:

$$P(\sum_{k=1}^{L} 4^k \phi^k c[i, i+k-1] < mpm_L)$$

Given that $c[i, i+k-1]$ is a $Binomial(n, p_{w_k})$, for large values of n it can be approximated as a $Normal(np_{w_k}, np_{w_k}(1-p_{w_k}))$. Also the sum can be approximated with

$$\sum_{k=1}^{L} 4^k \phi^k c[i, i+k-1] \rightarrow \mathcal{N}(\overline{\mu}, \overline{\sigma}^2)$$

where $\overline{\mu} = \sum_{k=1}^{L} 4^k \phi^k n p_{w_k} = n \sum_{k=1}^{L} \phi^k$ and $\overline{\sigma}^2 = \sum_{k=1}^{L} 4^k \phi^k n p_{w_k}(1-p_{w_k}) = n \sum_{k=1}^{L} \phi^k (1 - 1/4^k)$.

In practice the expected efficiency depends on the distribution of words in the string s, that will determine mpm_L. For example Figure 3 reports the number of nodes visited while computing max_L for all L for the string $TCGGCGGCAAC$. Similar results are obtained also for longer random sequences (data not shown). In general small values of ϕ drastically prune the tree.

4 Results

The Fast Entropic Profiler was tested in several DNA sequences, but in this section we report the results for two genomes. Here we illustrate an example of

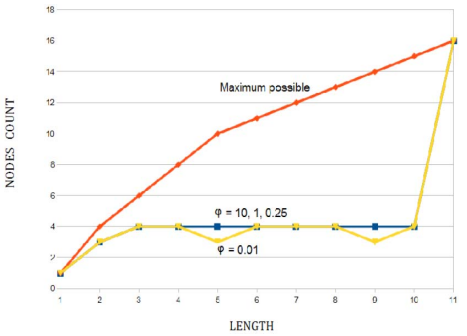


Fig. 3. Number of nodes visited for different values of ϕ while computing max_L for all possible L for the string TCGGCGGCAAC

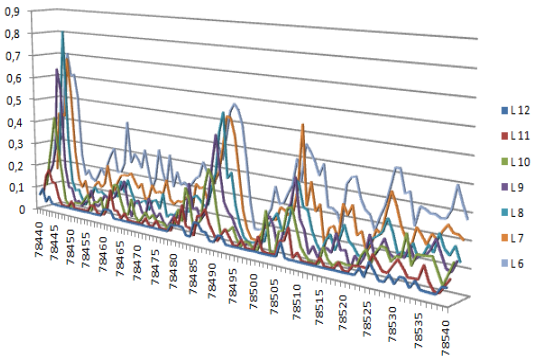


Fig. 4. Example of study of the E-Coli genome starting at position 78440 for various values of L

study around a target position. We can select a window length to study a certain range of values around the position. Also the length L can be chosen and in this case we search for pattern of length from 6 to 12. Note that after computing the values for $L = 12$ all other values for $L < 12$ can be computed in constant time. Figure 4 shows the output results for the Escherichia coli K12 genome with $\phi = 10$, starting position 78440 and window length of 100.

The figure reports the values of FastEP for all positions in the range [78440-78540]. For each position several values are reported varying the parameter L. The most important peak is at position 78445 and the value of L that maximizes this peak is $L = 8$. This highly rated motif is in fact GCTGGTGG, which corresponds to a Chi site, a region that modulates the activity of RecBCD (an enzyme involved in the chromosomal repair)[21]. It is important to notice that this pattern can be discovered just by looking at the histogram, and by analyzing the values L that maximize the score for this position, and without a previous knowledge of the length of the motif under study.

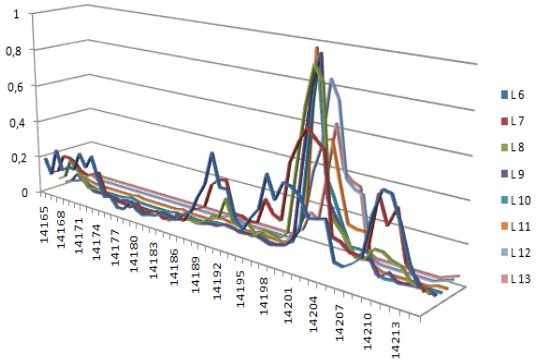


Fig. 5. Example of study of the H.Influenza genome starting at position 14165 for various values of L

Table 1. Running times in second for EP and FastEP

		FastEP		
Size	EP	Single Run	New Query	New Parameters
1 Mbases	12	4	0,09	1,5
1 Kbases	0,346	0,066	0,021	0,032

In Figure 5 a similar results is shown for the H.Influenza genome. We study the positions from 14165 to 14215 with $\phi = 10$ for various values of L. The most important peak is obtained at position 14202 for $L = 9$, that corresponds to the pattern AAGTGCGGT. This well known pattern represents an uptake signal sequence (USS+) involved in the horizontal gene transfer [13].

In a second series of experiments we test the time performance on a common laptop with a 1.5GHz Centrino and 2Gb of Ram. Table 1 reports the average times over 10 runs for two genomes of length 1kbases and 1Mbases. For all runs we use $L = 10$, $\phi = 10$ and a window of 100. In column "EP" is reported the time for the original method. For FastEP three times are illustrated. The construction and query correspond to the column "Single Run". A new query, e.g. a new starting position or a shorter L, is represented by the column "New Query". If a larger L or a new value of ϕ are required the inner structure is updated in a time reported in the last column. On a single run FastEP is always faster than the original method. If multiple queries are required the advantage becomes immediately embarrassing. The small space requirements and the improved performance will enable the study on large genomes.

Moreover in the original implementation the parameter L can not be greater than 15, whereas FastEP does not have limitation and can search for longer patterns.

5 Conclusions

To summarize we improve the original Entropic Profile with a faster and more flexible implementation that can search for longer patterns in a genome. We proposed a new normalization that can be efficiently computed within the inner structure of FastEP. We provide some examples where FastEP is used for the detection of conserved signals in a genome.

Acknowledgments. M. Comin was partially supported by the Ateneo Project CPDA110239. S. Mazzocca implemented the software FastEP.

References

1. Apostolico, A., Comin, M., Parida, L.: Varun: Discovering Extensible Motifs under Saturation Constraints. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(4), 752–762 (2010)
2. Apostolico, A., Comin, M., Parida, L.: Mining, compressing and classifying with extensible motifs. Algorithms for Molecular Biology 1, 4 (2006)
3. Apostolico, A., Comin, M., Parida, L.: Bridging Lossy and Lossless Compression by Motif Pattern Discovery. In: Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.) General Theory of Information Transfer and Combinatorics. LNCS, vol. 4123, pp. 793–813. Springer, Heidelberg (2006)
4. Apostolico, A., Comin, M., Parida, L.: Motifs in Ziv-Lempel-Welch Clef. In: Proceedings of IEEE DCC Data Compression Conference, pp. 72–81. Computer Society Press (2004)
5. Bernaola-Galván, P., Grosse, I., Carpena, P., Oliver, J., Román-Roldán, R., Stanley, H.: Finding Borders between Coding and Noncoding DNA Regions by an Entropic Segmentation Method. Physical Review Letters 85(6), 1342–1345
6. Comin, M., Parida, L.: Subtle motif discovery for the detection of DNA regulatory sites. In: Proceeding of Asia-Pacific Bioinformatics Conference, pp. 27–36 (2007)
7. Comin, M., Parida, L.: Detection of Subtle Variations as Consensus Motifs. Theoretical Computer Science 395(2-3), 158–170 (2008)
8. Comin, M., Verzotto, D.: Alignment-Free Phylogeny of Whole Genomes using Underlying Subwords. BMC Algorithms for Molecular Biology 7, 34 (2012)
9. Comin, M., Verzotto, D.: Whole-Genome Phylogeny by Virtue of Unic Subwords. In: Proceedings of 23rd International Workshop on Database and Expert Systems Applications, BIOKDD, pp. 190–194 (2012)
10. Comin, M., Verzotto, D.: The Irredundant Class Method for Remote Homology Detection of Protein Sequences. Journal of Computational Biology 18(12), 1819–1829 (2011)
11. Gene, Y., Burge, C.: Maximum Entropy Modeling of Short Sequence Motifs with Applications to RNA Splicing Signals. Journal of Computional Biology 11(2-3), 377–394 (2004)
12. Hagenauer, J., Dawy, Z., Gobel, B., Hanus, P., Mueller, J.: Genomic Analysis using Methods from Information Theory. In: Information Theory Workshop, pp. 55–59 (2004)

13. Karlin, S., Mrazek, J., Campbell, A.: Frequent oligonucleotides and peptides of the Haemophilus influenzae genome. Nucleic Acids Res. 24, 4263–4272 (1996)
14. Kurtz, S., Choudhuri, J., Ohlebusch, E., Schleiermacher, C., Stoye, J., Giegerich, R.: Reputer: The manifold applications of repeat analysis on a genome scale. Nucleic Acids Res. 29(22), 4633–4642 (2001)
15. McCreight, E.M.: A space-economical suffix tree construction algorithm. Journal of ACM 23, 262–272 (1976)
16. Meek, C., Patel, J., Kasetty, S.: Oasis: An online and accurate technique for local-alignment searches on biological sequences. In: Proceedings of 29th International Conference on Very Large Databases, pp. 910–921 (2003)
17. Menconi, G., Marangoni, R.: A compression-based approach for coding sequences identification. I. Application to prokaryotic genomes. J. Comput Biol. 13(8), 1477–1488 (2006)
18. Nalla, V., Rogan, P.: Automated Splicing Mutation Analysis by Information Theory. Human Mutaion 25, 334–342 (2005)
19. Schneider, T., Stormo, G., Gold, L., Ehrenfeucht, A.: Information content of binding sites on nucleotide sequences. Journal of Molecular Biology 188, 415–431 (1986)
20. Shannon, C.: A Mathematical Theory of Communication. Bell System Technical Journal 27(3), 379–423 (1948)
21. Sourice, S., Biaudet, V., El Karoui, M., Ehrlich, S.D., Gruss, A.: Identification of the Chi site of Haemophilus influenzae as several sequences related to the Escherichia coli Chi site. Mol. Microbiol. 27, 1021–1029 (1998)
22. Ukkonen, E.: On-line construction of suffix trees. Algorithmica 14(3), 249–260 (1995)
23. Vinga, S., Almeida, J.S.: Local Rényi entropic profiles of DNA sequences. BMC Bioinformatics 8, 393 (2007)
24. Yockey, H.: Origin of life on earth and Shannon's theory of communication. Comput. Chem. 24(1), 105–123 (2000)
25. Waterman, M.S.: An Introduction to Computational Biology: Maps, Sequences and Genomes. Chapman Hall (1995)
26. Ziv, J., Lempel, A.: A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23(3), 337–343 (1977)

Author Index