

Eleazar Eskin
Trey Ideker
Ben Raphael
Christopher Workman (Eds.)

LNBI 4023

Systems Biology and Regulatory Genomics

Joint Annual RECOMB 2005 Satellite Workshops
on Systems Biology and on Regulatory Genomics
San Diego, CA, USA, December 2005, Revised Selected Papers

 Springer

Lecture Notes in Bioinformatics

4023

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

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

Subseries of Lecture Notes in Computer Science

Eleazar Eskin Trey Ideker
Ben Raphael Christopher Workman (Eds.)

Systems Biology and Regulatory Genomics

Joint Annual RECOMB 2005 Satellite Workshops
on Systems Biology and on Regulatory Genomics
San Diego, CA, USA, December 2-4, 2005
Revised Selected Papers

Series Editors

Sorin Istrail, Brown University, Providence, RI, USA

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

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

Volume Editors

Eleazar Eskin

Ben Raphael

University of California

Department of Computer Science

San Diego, CA, USA

E-mail: {eeskin,braphael}@cs.ucsd.edu

Trey Ideker

Christopher Workman

University of California

Department of Bioengineering

San Diego, CA, USA

E-mail: {trey,cworkman}@bioeng.ucsd.edu

Library of Congress Control Number: 2006940071

CR Subject Classification (1998): F.2, G.3, E.1, H.2.8, J.3

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN 0302-9743

ISBN-10 3-540-48293-8 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-48293-2 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007

Printed in Germany

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

Printed on acid-free paper SPIN: 11916338 06/3142 5 4 3 2 1 0

Preface

It has become increasingly evident that the use of large-scale experimental data and the application of principles from systems biology are gaining widespread acceptance in mainstream biology. Systems biology involves the use of global cellular measurements, i.e., genomic, proteomic, and metabolomic, to construct computational models of cellular processes and disease. These approaches involve an integration of experimental and computational techniques and may include: 1) developing models of cellular processes, 2) measuring the response to perturbations of model components, and 3) iteratively formulating and testing new hypotheses for unexpected observations.

A research area that is particularly important to systems biology is the study of gene regulatory networks. Although genome sequencing efforts have been tremendously successful, much is unknown about the regulation of these sequenced genomes. Automatic methods for helping decipher the regulatory mechanism are crucial for understanding the regulatory network as a whole. However, many new challenges are presented when analyzing complete genomes. These challenges include motif discovery in large genomes, leveraging information from multiple genomes, detection of weak signals and incorporating different types of genomic data such as protein localization data and gene expression. Novel methodology will be particularly relevant given the hypothesis that the observed phenotypic differences between organisms with very similar genomes may be due to variations in the gene regulation.

The amount of research in both of these areas has exploded in recent years, as witnessed by the number of research presentations at meetings such as RECOMB, ISMB, PSB, the Biopathways Consortium, and ICSB. The jointly held RECOMB Satellite on Systems Biology and the RECOMB Satellite on Regulatory Genomics provide a forum for addressing these challenges.

This year's workshops also included a special session on Computational Developmental Biology organized by David Gifford. Unique challenges posted by developmental biology include: 1) computational model representations that can express the execution of complex natural programs over time, 2) the identification of key developmental state variables and experimental methods to reliably observe these variables, and 3) the use of computational models to understand developmentally related disease, and to help develop therapeutics including the programming of stem cells as therapeutic agents. Invited speakers at this year's special session addressed the question: "What key developmental biology problems can now be examined from a systems biology perspective, and what data are necessary to do so?". The goal of this special session was to help computational and systems biologists understand both the challenge and excitement of working on development.

Organization

Steering Committee: Regulatory Genomics

Pierre Baldi	University of California, Irvine
Michael Eisen	Lawrence Berkeley National Lab
Eleazar Eskin	University of California, San Diego
Pavel Pevzner	University of California, San Diego

Steering Committee: Systems Biology

Leroy Hood	Institute for Systems Biology
Trey Ideker	University of California, San Diego
Douglas Lauffenburger	MIT
Satoru Miyano	University of Tokyo
Ron Shamir	Tel Aviv University

Organizing Committee

Eleazar Eskin	University of California, San Diego
David Gifford	Massachusetts Institute of Technology
Trey Ideker	University of California, San Diego
Teresa M. Przytycka	NIH/NLM/NCBI
Ben Raphael	University of California, San Diego
Cenk Sahinalp	Simon Fraser University
Samantha Smeraglia	University of California, San Diego
Christopher Workman	University of California, San Diego

Program Committee

John Aitchison	Institute for Systems Biology
Adam Arkin	University of California, Berkeley
Gary Bader	Memorial Sloan-Kettering Cancer Center
Pierre Baldi	University of California, Irvine
Yoseph Barash	Hebrew University
Mathieu Blanchette	McGill University
Gal Chechik	Stanford University
Francis Chin	University of Hong Kong
Michael Cusick	Harvard Medical School
Eric Davidson	CalTech
Michael Eisen	Lawrence Berkeley National Lab
Eleazar Eskin	University of California, San Diego

Nir Friedman	Hebrew University
Irit Gat-Vicks	Tel Aviv University
Mikhail Gelfand	Moscow State University
David Gifford	Massachusetts Institute of Technology
Sridhar Hannenhalli	University of Pennsylvania
Jeff Hasty	University of California, San Diego
Leong Hon Wai	National University of Singapore
Leroy Hood	Institute for Systems Biology
Trey Ideker	University of California, San Diego
Richard Karp	University of California, Berkeley
Uri Keich	Cornell University
Douglas Lauffenburger	Massachusetts Institute of Technology
Christina Leslie	Columbia University
Mike Levine	University of California, Berkeley
Hao Li	University of California, San Francisco
Nick Luscombe	European Bioinformatics Institute
Satoru Miyano	University of Tokyo
Dana Pe'er	Harvard Medical School
Pavel Pevzner	University of California, San Diego
Tzachi Pilpel	Weizmann Institute of Science
Teresa M. Przytycka	NIH/NLM/NCBI
Ben Raphael	University of California San Diego
Bing Ren	University of California, San Diego
Aviv Regev	Harvard Medical School
Mireille Regnier	INRIA
Marie-France Sagot	INRIA
Cenk Sahinalp	Simon Fraser University
Eran Segal	Rockefeller University
Ron Shamir	Tel Aviv University
Roded Sharan	Tel Aviv University
Amos Tanay	Tel Aviv University
Alfonso Valencia	Centro Nacional de Biotecnologia
Marc Vidal	Harvard Medical School
Christopher Workman	University of California, San Diego
Eric Xing	Carnegie Mellon University
Zohar Yakhini	Agilent
Ralf Zimmer	LMU, Institut für Informatik

Sponsoring Institutions

Industry-University Cooperative Research Program, The UC Discovery Grant
California Institute for Telecommunications and Information Technology, Cal-(IT)²

Table of Contents

An Interactive Map of Regulatory Networks of <i>Pseudomonas aeruginosa</i> Genome	1
<i>Weihui Wu, Yongling Song, Shouguang Jin, and Su-Shing Chen</i>	
The Pathalyzer: A Tool for Analysis of Signal Transduction Pathways	11
<i>David L. Dill, Merrill A. Knapp, Pamela Gage, Carolyn Talcott, Keith Laderoute, and Patrick Lincoln</i>	
Decomposition of Overlapping Protein Complexes: A Graph Theoretical Method for Analyzing Static and Dynamic Protein Associations	23
<i>Elena Zotenko, Katia S. Guimarães, Raja Jothi, and Teresa M. Przytycka</i>	
Comparison of Protein-Protein Interaction Confidence Assignment Schemes	39
<i>Silpa Suthram, Tomer Shlomi, Eytan Ruppin, Roded Sharan, and Trey Ideker</i>	
Characterization of the Effects of TF Binding Site Variations on Gene Expression Towards Predicting the Functional Outcomes of Regulatory SNPs	51
<i>Michal Lapidot and Yitzhak Pilpel</i>	
Comparative Systems Biology of the Sporulation Initiation Network in Prokaryotes	62
<i>Michiel de Hoon and Dennis Vitkup</i>	
Improvement of Computing Times in Boolean Networks Using Chi-square Tests	70
<i>Haseong Kim, Jae K. Lee, and Taesung Park</i>	
Build a Dictionary, Learn a Grammar, Decipher Stegoscripts, and Discover Genomic Regulatory Elements	80
<i>Guandong Wang and Weixiong Zhang</i>	
Causal Inference of Regulator-Target Pairs by Gene Mapping of Expression Phenotypes	95
<i>David Kulp and Manjunatha Jagalur</i>	
Examination of the tRNA Adaptation Index as a Predictor of Protein Expression Levels	107
<i>Orna Man, Joel L. Sussman, and Yitzhak Pilpel</i>	

Improved Duplication Models for Proteome Network Evolution	119
<i>Gürkan Bebek, Petra Berenbrink, Colin Cooper, Tom Friedetzky, Joseph H. Nadeau, and S. Cenk Sahinalp</i>	
Application of Expectation Maximization Clustering to Transcription Factor Binding Positions for cDNA Microarray Analysis	138
<i>Chih-Yu Chen, Von-Wun Soo, and Chi-Li Kuo</i>	
Combinatorial Genetic Regulatory Network Analysis Tools for High Throughput Transcriptomic Data	150
<i>Elissa J. Chesler and Michael A. Langston</i>	
Topological Robustness of the Protein-Protein Interaction Networks	166
<i>Chien-Hung Huang, Jywe-Fei Fang, Jeffrey J.P. Tsai, and Ka-Lok Ng</i>	
A Bayesian Approach for Integrating Transcription Regulation and Gene Expression: Application to <i>Saccharomyces Cerevisiae</i> Cell Cycle Data	178
<i>Sudhakar Jonnalagadda and Rajagopalan Srinivasan</i>	
Probabilistic <i>in Silico</i> Prediction of Protein-Peptide Interactions	188
<i>Wolfgang Lehrach, Dirk Husmeier, and Christopher K.I. Williams</i>	
Improved Pattern-Driven Algorithms for Motif Finding in DNA Sequences	198
<i>Sing-Hoi Sze and Xiaoyan Zhao</i>	
Annotation of Promoter Regions in Microbial Genomes Based on DNA Structural and Sequence Properties	212
<i>Huiquan Wang and Craig J. Benham</i>	
An Interaction-Dependent Model for Transcription Factor Binding	225
<i>Li-San Wang, Shane T. Jensen, and Sridhar Hannenhalli</i>	
Computational Characterization and Identification of Core Promoters of MicroRNA Genes in <i>C. elegans</i> , <i>H. sapiens</i> and <i>A. thaliana</i>	235
<i>Xuefeng Zhou, Jianhua Ruan, Guandong Wang, and Weixiong Zhang</i>	
A Comprehensive Kinetic Model of the Exocytotic Process: Evaluation of the Reaction Mechanism	249
<i>Aviv Mezer, Eran Bosis, Uri Ashery, Esther Nachliel, and Menachem Gutman</i>	
Author Index	259

An Interactive Map of Regulatory Networks of *Pseudomonas aeruginosa* Genome

Weihui Wu¹, Yongling Song², Shouguang Jin¹, and Su-Shing Chen²

¹Department of Molecular Genetics and Microbiology,
University of Florida, Gainesville, FL 32611
weihuiwu@ufl.edu, sjin@mgm.ufl.edu

²Department of Computer Information Science and Engineering,
University of Florida, Gainesville, FL 32611
{yosong, suchen}@cise.ufl.edu

Abstract. For studying gene regulatory and protein signaling networks, we have developed an interactive map for the *Pseudomonas aeruginosa* genome. We first represent genes, proteins and their regulatory networks in a relational database. Known regulatory networks of the genome in the PubMed literatures are extracted by a manual and later a semi-automated text-mining method. Then a graphical interface displays these networks upon the query of specific genes, proteins or subsystems (i.e., groups of genes or proteins) on these networks. The interactive map has another capability of browsing those networks. The method can be extended to any other genome. Our objective is to develop this interactive map for the *Pseudomonas aeruginosa* community so that new research results may be ingested into the database, while annotations may be developed incrementally on the existing regulatory elements. Eventually some standards might be necessary for a long-term modeling and compilation of regulatory networks.

1 Introduction

With the exponential growth of available genomic sequences, various bioinformatics tools became available for comparative genomic analysis to annotate gene functions and build metabolic pathways (4, 7). However, establishing a functional regulatory network still relies largely on experimental approaches. Understanding the relationships among various biological “subsystems” enable us to understand the real biology at organism level, making bioengineering as well as drug discovery easier. *Pseudomonas aeruginosa* is an environmental bacterium, which causes serious human infections, especially those with reduced immunity, patients with Cystic Fibrosis or severe burns (2, 9). A large number of virulence genes and regulatory genes encoded by this organism make this bacterium one of the most successful pathogen on the earth. A complicated regulatory network coordinates the expression of various virulence genes as well as different functional groups of genes for an efficient host infection and survival in hostile host environments (5, 13). Prolonged treatments with antibiotics often result in multi-drug resistant isolates, which eventually cause death in the infected

individuals (11). Therefore, there is an urgent need to develop new antimicrobial strategies for an effective control of this deadly bacterium.

Through decades of active research, tremendous amount of experimental data are available on the gene function and their regulation in *P. aeruginosa* (3, 8). However, these information are embedded in tens of thousands published literatures, thus difficult for individual researchers to extract the information for a comprehensive view of the field. Also, as more and more new research data become available, it is difficult for individual scientists to keep up with all the new information. There is a need to develop an interactive database which not only compiles experimental evidences but also logically integrate the knowledge related to gene function and regulation in *P. aeruginosa*.

The whole genome sequence of this microorganism has been completed several years ago and freely available to the public (12). The complete sequence of the genome was the largest bacterial genome sequenced to date when published, with 6.3-Mbp in size encoding 5570 predicted genes (12). Functions of only 480 of those encoded proteins have been demonstrated experimentally while the rest, including 1059 where functions of strongly homologous genes have been demonstrated experimentally in other organisms, 1524 whose functions were proposed based on the presence of conserved amino acid motif, structural feature or limited homology, and 2507 which are homologs of previously reported genes of unknown function, or no homology to any previously reported sequences. Most interestingly, as high as 8% of the genome encodes transcriptional regulators, which is consistent with the observed bacterial adaptability to various growth environments through alteration of gene expression pattern (6, 12).

2 The Knowledge Base of *P. aeruginosa*

In the current project in progress, we intend to collect phenotypes of *P. aeruginosa* mutants and construct an interactive map of regulatory networks based on published literatures. A regulatory network will show relationships among genes, operons, regulons and stimulons. This regulatory network map will help researchers have a global view on the function of one gene and the relationship among several regulatory elements, facilitating the acquisition of relative information and design of future experiments. The database will be searchable by gene names or PA numbers, which have links to provide the following information:

- (1) Mutant phenotypes, which include genotypes and phenotypes of the mutants and the parent strains as well as published references .
- (2) Gene regulation at each subsystem level.
- (3) Relationship among subsystems.

A regulatory network will include the genes if they belong to certain regulons/stimulons and related signals for activation/repression. Regulation at the transcriptional level (activation or repression), posttranscriptional (mRNA stability), translational or post-translational (protein-protein interaction/modification) will be indicated. As a reference for data reliability, the evidence for regulation – whether it

was based on genetic evidence, biochemical tests or sequence-based prediction - will also be included. Using functional subsystem as a unit, the relative position of each gene in the regulatory cascade will be placed in the map. Also, placing regulatory genes in center, their regulatory role on various subsystems will also be marked.

In order to achieve the objectives, we have extracted information from 150 published papers in a systematic manner, which can be automated or semi-automated by natural language processing techniques. Phenotypes of all the mutants and references have been recorded. A database of gene regulatory networks has been developed, which is the basis for the computer-generated interactive maps. In this work, we have collected the virulence genes relevant to human infections, including the type IV pili subsystem involved in adhesion and twitching motility (10) and flagellar subsystem for the bacterial motility (1). Each subsystem involves tens of genes and is tightly and coordinately regulated. In existing literatures, some gene regulatory networks have been drawn manually, such as Flagella (1). However our database provides not only detailed gene relationships within subsystems, but also relationships between different subsystems so that users will have a global view of gene regulatory networks, and find specific potential connections between subsystems.

Upon completion, this interactive database will be released to the public. The interactive system may be further improved and updated by the research community. Our long-term goals are to build a database on the gene regulation for a comprehensive view of the regulatory networks at the genomic level; to automate the data extraction from published literatures; and to automate regulatory network building tools for various organisms based on this modeling methodology.

3 Modeling of Regulatory Elements

3.1 System Architecture

The modeling consists of a database and a graphical tool. The graphical tool will be interoperable with the web services so that users can search and visualize during any session. The database stores all necessary knowledge about regulatory networks. The database schema has 4 tables. The System Table presents subsystems and their relationships. The Gene Table includes all basic information about genes and proteins, which will be nodes in the regulatory networks, while subsystems may be either nodes or regulatory networks. The Edge Table describes the edge information, including the attributes: nodes, directions, types, relationships and subsystems. Directions are Active or Repressive - represented as positive and negative - and Be Activated and be Repressed. Types indicate DNA, RNA, and Protein Binding, Signal Molecule Production, Signal Sensing, and Signal/Molecule Binding. The Gene Information Table includes all other information about genes, such as references, genotypes, strains, phenotypes, and comments. The following figure illustrates a very small portion of the graph. Various annotations and

symbols are introduced to model the regulatory networks. The other two figures – Figures 3 and 4 – can be zoomed out to above 400% to visualize more details of regulatory networks.

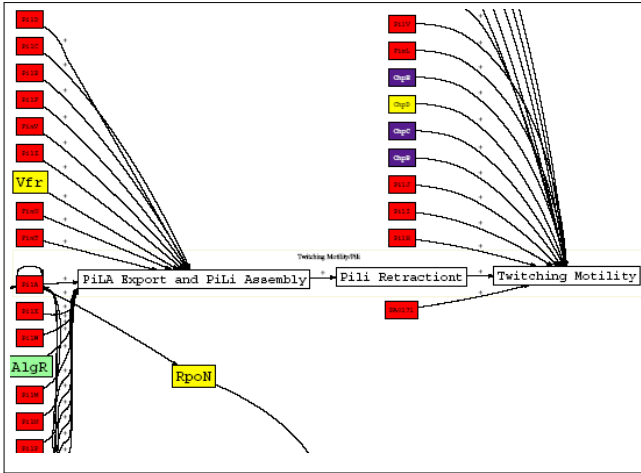


Fig. 1. A Glimpse of the Pili subsystem

The system architecture includes three parts: Data Collection and Management System (DCM), Dynamic Graph Generation System (DGG) and Web-based User Navigation System (WUN). The DCM system will collect the regulatory network knowledge from the PubMed database, extract and represent regular elements into a spreadsheet table. Then by using a parser program, we insert the data in the spreadsheet table into our *Pseudomonas aeruginosa* Database (PADB) server. When users access the WUN system, they can query some special requests, for example: one subsystem graph, the whole system graph, or other graphs based on previous search results. The WUN system sends users' requests to the WWW server. The WWW server sends users' requests to the DGG system. DGG will do three things: first, it will analyze the users' requests and generate some SQL commands based on users' requests. Then it will pass the SQL to the PADB database server, the database server will execute the SQL commands and return the resulting data to the DGG system. DGG will analyze the data and generate text file with the DOT language format. Then DGG will call the Graphviz software, which will generate the graph visualization based on the DOT file and output the graph into a PDF file. Then DGG will pass the PDF file into WUN. Now a user can view the requested graph of a regulatory network. From any displayed regulatory network, a user can further click on specific genes or other elements to search for other regulatory networks. The system architecture is depicted in the following figure:

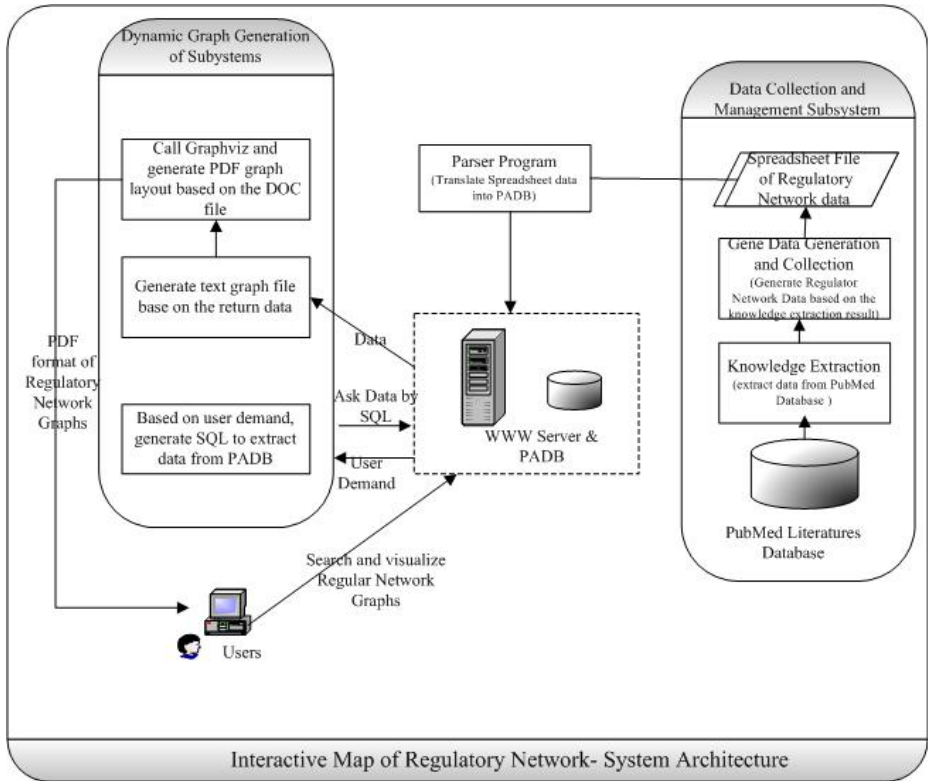


Fig. 2. System Architecture of Interactive Map of Regulatory Networks

3.2 Automated Natural Language Processing for Information Extraction

We have developed natural language methods for text mining of MEDLINE or PubMed databases (15). For this work, we are developing automated natural language processing techniques to extract two kinds of information: phenotypes of mutants and relationships between genes from literatures in the PubMed database. The basic techniques are noun phrases extraction and listing of noun phrases of special headings. First, we need to find out how many mutants have been studied in a paper on *Pseudomonas aeruginosa*. Usually, there is a table in the paper showing all the strains used in the study and their genotypes. If such a table is not provided, we look for information about phenotypes of mutants in the papers, such as

- “a mutation of A gene is constructed in one strain”*,
- “a gene is knocked out in one stain”*,
- “a gene mutant shows...”*.
- “the X phenotype of A gene mutant is...”*.
- “compared to wild type strain”*,
- “we assayed the X phenotype in ...”*.

After these phrases, we will find the genotypes usually. To extract information about relationships between genes or one gene to one subsystem, we look for information, such as

“A gene activates the expression of B gene”

“A gene or protein+ description of relationship+ gene or subsystem”

The key in information extraction is that whenever one gene or protein is mentioned in the text, the description of its function or relationship with other gene/subsystem may be somewhere close.

4 *Pseudomonas aeruginosa* Subsystems and Their Significance

So far, the interactive map of gene regulation networks of *Pseudomonas aeruginosa* contains eight subsystems: flagellum, pili, Type III secretion, Iron acquisition, quorum sensing, biofilm, alginate synthesis and multi-drug efflux. These subsystems compose the overall regulatory networks of *P. aeruginosa*. Among them, there is interaction between genes in different subsystems. In this abstract, we only demonstrate the first two subsystems. As we are constructing the database, more subsystems will be extracted and integrated. The overview of these two subsystems are in the Figure 3. Figure 4 gives the search result page when the user search for the gene “RpoN”. The interactive map consists of several layers, the top view is the global view of all subsystems, while the lower layers display zoom-in and zoom-out a subsystem map.

Flagellum serves as a motive organelle on the surface of bacterium. The flagellum consists of basal body, hook, flagellar filament and motor. The basal body anchors the flagellum on the surface of the bacterium; the hook functions as a joint, connecting the filament to the basal body. The filament functions as a propeller and the motor generates the rotation of the flagellum. By rotating flagellum, bacteria can move in the surrounding environment. Two types of movement depend on flagella, swimming and swarming. Swimming is a movement of bacteria in the surround liquid and swarming is a surface translocation by groups of bacteria. Besides flagellum, *P. aeruginosa* produces another motive organelle named type IV pilus. Pilus is a polar filament structure, mediating attachment to host epithelial cells and one type of surface translocation named twitching motility. The pilus is composed of a small subunit (pilin). Pilin is synthesized in the cytoplasm as pre-pilin and translocated through inner membrane, cell wall and outer membrane to the surface of bacterium. During translocation, pre-pilin is cleaved to pilin.

The pilus is able to extend and retract, resulting in the surface translocation (twitching motility). The Type III secretion system (TTSS) is a potent virulence factor possessed by *P. aeruginosa*. The TTSS contains a syringe like apparatus, which can directly inject effector proteins from bacterium cytoplasm into host cell cytosol, causing cell death. Four effector proteins have been found in *P. aeruginosa*, ExoS, ExoT, ExoY, ExoU. Expression and secretion of the TTSS regulon can be stimulated by a direct contact with the host cell or by growth under low Ca²⁺ environment. Iron is essential for the metabolism and survival of *P. aeruginosa*. To acquire iron from surrounding environment *P. aeruginosa* produces and secretes iron-chelating compound, named siderophore. Two types of siderophores, pyoverdine and

pyochelin, are produced by *P. aeruginosa*. The pyoverdine and pyochelin synthesis genes and receptors are under the negative control of a regulator, Fur and under iron deplete environments, the expression of these genes are derepressed. *P. aeruginosa* involves a signaling system for cell-cell communication, named quorum sensing. *P. aeruginosa* possesses three quorum sensing systems, known as las, rhl and PQS (pseudomonas quinolone signal). Each system contains a small molecule involved in signal communication. The las and rhl systems use acyl-homoserine lactones, C4-HSL and 3OC12-HSL as signal molecules, respectively. The signal molecule of PQS system is quinolone. The signal molecules are secreted into the surrounding environment and when their concentrations reach a threshold, they can interact with their own receptors and change the gene expression in other cells. The three quorum sensing systems can interact with each other. In *P. aeruginosa* many genes, including virulence genes are under the control of quorum sensing systems.

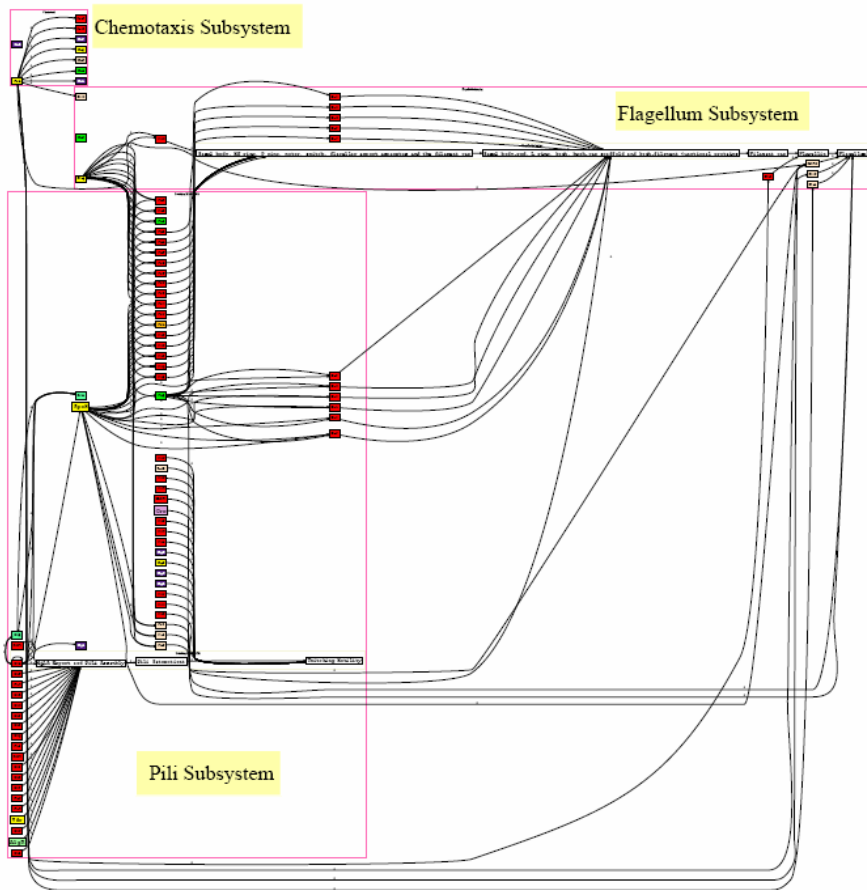


Fig. 3. Three Subsystems in Regulatory Networks of *Pseudomonas Aeruginosa*

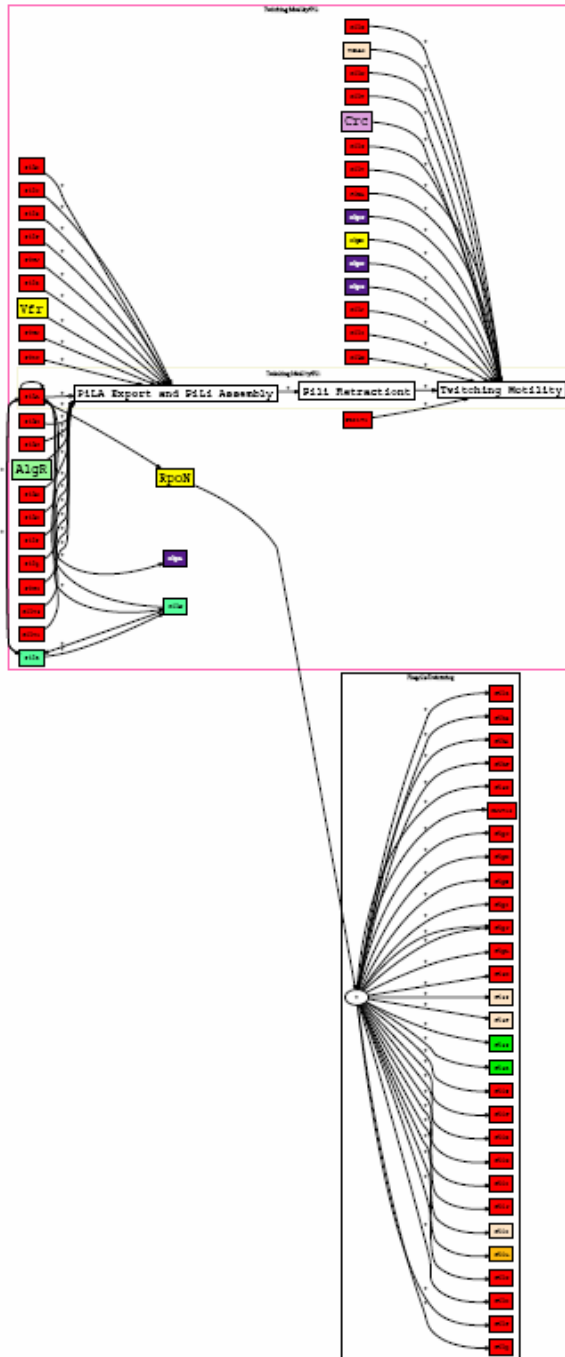


Fig. 4. The Search Result of Gene “RpoN” in the Regulatory Networks

During chronic infection of cystic fibrosis (CF) patient's airway, *P. aeruginosa* forms biofilm. The biofilm is composed of bacteria microcolonies surrounded by exopolysaccharide. The formation of biofilm requires flagella, pili, exopolysaccharide and quorum sensing systems. In biofilm, *P. aeruginosa* is highly resistant to host immune system and antibiotics. Most *P. aeruginosa* clinical isolates from CF patients display a mucoid phenotype. The mucoidy is caused by over production of an exopolysaccharide, alginate. The production of alginate is repressed by an inner membrane protein: MucA. A high proportion of clinical isolates from CF patients has been found to contain mutation in mucA gene, resulted in the over production of alginate. *P. aeruginosa* is highly resistant to multiple antimicrobial agents. One mechanism of its intrinsic resistance is chromosomally encoded multi-drug efflux system. The multi-drug efflux system contains three components, an inner membrane drug transporter, a channel-forming outer membrane protein and a periplasm protein connecting the other two. The multi-drug efflux system can pump out antibiotics from cytoplasm and periplasm. *P. aeruginosa* possesses several multi-drug efflux systems, with different substrate specificity.

5 Conclusion

This work in progress builds an interactive map of regulatory networks for the *Pseudomonas aeruginosa* Genome. We have developed a semi-automated technique to extract information about regulatory networks from the PubMed database. We are still working on fully automated methods developed in our earlier work (16).

References

1. Dasgupta, N., M. C. Wolfgang, A. L. Goodman, S. K. Arora, J. Jyot, S. Lory, and R. Ramphal. A four-tiered transcriptional regulatory circuit controls flagellar biogenesis in *Pseudomonas aeruginosa*. *Mol Microbiol* 50, (2003) 809-24.
2. Donaldson, S. H., and R. C. Boucher. Update on pathogenesis of cystic fibrosis lung disease. *Curr Opin Pulm Med* 9, (2003) 486-91.
3. Erwin, A. L., and D. R. VanDevanter. The *Pseudomonas aeruginosa* genome: how do we use it to develop strategies for the treatment of patients with cystic fibrosis and *Pseudomonas* infections? *Curr Opin Pulm Med* 8, (2002) 547-51.
4. Gibson, G., and E. Honeycutt. The evolution of developmental regulatory pathways. *Curr Opin Genet Dev* 12, (2002) 695-700.
5. Goodman, A. L., and S. Lory. Analysis of regulatory networks in *Pseudomonas aeruginosa* by genomewide transcriptional profiling. *Curr Opin Microbiol* 7, (2004) 39-44.
6. Greenberg, E. P. 2000. Bacterial genomics. Pump up the versatility. *Nature* 406:947-8.
7. Lange, B. M., and M. Ghassemian. Comprehensive post-genomic data analysis approaches integrating biochemical pathway maps. *Phytochemistry* 66, (2005) 413-51.
8. Larbig, K., C. Kiewitz, and B. Tummler. Pathogenicity islands and PAI-like structures in *Pseudomonas* species. *Curr Top Microbiol Immunol* 264, (2002) 201-11.
9. Lyczak, J. B., C. L. Cannon, and G. B. Pier. Establishment of *Pseudomonas aeruginosa* infection: lessons from a versatile opportunist. *Microbes and Infection* 2 (2000) 1051-1060.

10. Mattick, J. S. 2002. Type IV pili and twitching motility. *Annu Rev Microbiol* 56. (2000) 289-314.
11. Rossolini, G. M., and E. Mantengoli. Treatment and control of severe infections caused by multiresistant *Pseudomonas aeruginosa*. *Clin Microbiol Infect* 11 Suppl 4. (2005)17-32.
12. Stover, C. K., X. Q. Pham, A. L. Erwin, S. D. Mizoguchi, P. Warrenner, M. J. Hickey, F. S. Brinkman, W. O. Hufnagle, D. J. Kowalik, M. Lagrou, R. L. Garber, L. Goltry, E. Tolentino, S. Westbrook-Wadman, Y. Yuan, L. L. Brody, S. N. Coulter, K. R. Folger, A. Kas, K. Larbig, R. Lim, K. Smith, D. Spencer, G. K. Wong, Z. Wu, I. T. Paulsen, J. Reizer, M. H. Saier, R. E. Hancock, S. Lory, and M. V. Olson. Complete genome sequence of *Pseudomonas aeruginosa* PA01, an opportunistic pathogen. *Nature* 406. (2000) 959-64.
13. Woods, D. E. 2004. Comparative genomic analysis of *Pseudomonas aeruginosa* virulence. *Trends Microbiol* 12. (2004) 437-9.
14. Ellson, J., E.R Gansner, L. Koutsofios, S.C. North, and G. Woodhull. Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools, chapter in *Graph Drawing Software*, M. Jugner and P. Mutzels, eds., Springer-Verlag. (2003)
15. Kim, H. and Chen, S. Ontology search and text mining of MEDLINE database, *DATA MINING IN BIOMEDICINE*, edited by P.M. Pardalos et al, Springer, to appear.(2005)

Bibliography

Weihui Wu is a graduate student of MGM and Yongling Song is a graduate student of CISE. Shouguang Jin is a professor of MGM with PhD from the University of Washington, and Su-Shing Chen (email:suchen@cise.ufl.edu) is a professor of CISE with PhD from the University of Maryland. They are also members of the Systems Biology Lab at the UF Genetics Institute.

The Pathalyzer: A Tool for Analysis of Signal Transduction Pathways

David L. Dill¹, Merrill A. Knapp², Pamela Gage³,
Carolyn Talcott², Keith Laderoute², and Patrick Lincoln²

¹ Stanford University

² SRI International

³ Highwire Press

Abstract. The Pathalyzer is a program for analyzing large-scale signal transduction networks. Reactions and their substrates and products are represented as transitions and places in a safe Petri net. The user can interactively specify goal states, such as activation of a particular protein in a particular cell site, and the system will automatically find and display a pathway that results in the goal state – if possible. The user can also require that the pathway be generated without using certain proteins. The system can also find all individual places and all pairs of places which, if knocked out, would prevent the goals from being achieved. The tool is intended to be used by biologists with no significant understanding of Petri nets or any of the other concepts used in the implementation.

1 Introduction

Signal transduction pathways in eukaryotic cells relay information received in the form of physical or chemical stimuli from both the external and internal environment to various intracellular targets. The most common targets are found in the active genome, where subsequent changes in specific gene expression occur in response to the original signal(s). These pathways consist of sequences of biochemical processes that transduce a signal usually through the modification and translocation of proteins and other molecules. Signaling pathways are embedded in large networks having multiple interactions, leading to phenomena such as redundancy and cross talk.

To someone with a computer design background, signaling pathways appear to have some properties in common with hardware and software control mechanisms. For example, pathways can show conditional activation, allowing different or variable responses to different combinations of signals. Currently, much work in signal transduction biology is focused on understanding molecular details of individual pathways in signaling networks. However, as genomic, proteomic, and other high order experimental information accumulates, signaling and other biological pathways will be increasingly represented within complex biological networks. Therefore, it will be essential to acquire at least a qualitative understanding of the global properties of these networks using computational tools. For example, it would be of value to be able to compute an answer to the general question “How are signaling pathways perturbed when environmental conditions change, or when a biochemical process is disrupted?”

This paper describes a prototype software tool, called the Pathalyzer, for querying large-scale qualitative information about signaling pathways. This tool, which uses

Petri nets to represent signaling pathways, is unique in several ways. It has a high-level Boolean approach to modeling, based on the Pathway Logic system [319], that enables the analysis of large biological networks. The analysis can answer questions, of types commonly asked by bench biologists, that require searching *all possible pathways* consisting of specific signaling molecules and their associated reactions. Thus, *the model has no built-in concept of pathways*; instead, pathways are generated dynamically from a collection of rules describing individual biochemical steps in signal transduction in response to queries from a user. This capability allows the Pathalyzer to generate a very large number of alternative pathways under the control of a user, who can display differences between the generated pathways. The tool uses sophisticated algorithms developed within the Petri net community for solving the computationally difficult *reachability problem* – in real time. The user needs to understand very little about Petri nets or the various algorithms used in the tool, because the tool provides an easy-to-use interface for entering queries and displaying the results.

A model of a mammalian signaling network has been developed consisting of hundreds of signaling and other molecules involved in the epidermal growth factor receptor (EGFR) system. Cancer biologists are strongly interested in EGFR system, so it has been extensively studied experimentally as a paradigm for growth factor effects on the proliferation and survival of diverse mammalian cells. All of the information incorporated within the model was curated from the appropriate scientific literature. For this model, under development as part of the Pathway Logic project [3418], the analysis algorithms of the Pathalyzer run in real time and produce results consistent with the published literature.

Briefly, in Pathway Logic biochemical processes are represented as rules written in Maude [12], a system based on rewriting logic [1314]. The Petri nets used by the Pathalyzer are extracted automatically from these Maude rules, and displayed for analysis as described below. The models used by the Pathalyzer are not specific to Maude. It would be possible to generate them from SBML [7] models or to enter the models via a web form interface. Using an existing model requires no knowledge of the model data entry process.

An interactive demonstration of the pathway logic viewer is available at <http://mcs.une.edu.au/~iop/Bionet/>.

2 Modeling Pathways as Petrinets

2.1 Petri Nets

A Petri net is a graphical representation of possible system behavior that generalizes the idea of a state machine to make notions of concurrency and conflict explicit in the structure of the graph. Also, Petri nets have precise, mathematically defined behavior, and sophisticated analysis techniques have been developed over several decades.

A Petri net is a directed bipartite graph with two types of vertices, called *places* and *transitions* (see Fig. 1). Places are drawn as circles, and transitions are drawn as boxes or thick bars. Each transition has a set of *input places* and a set of *output places*; similarly, each place has input and output transitions. There are directed edges (often

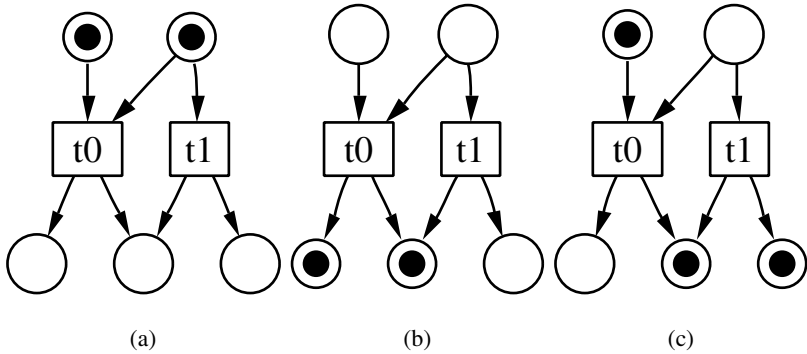


Fig. 1. A basic Petri net example

called “arcs”), drawn as arrows, from the input places to the transition and from the transition to its output places.

The behavior of a system is captured by moving markers called *tokens* around on the net according to certain rules. In Fig. 1, the tokens are shown in the conventional way as smaller black circles inside the places. Mathematically, a placement of tokens on a Petri net is a *marking*, which is an assignment of a non-negative number of tokens (possibly 0) to each place. A place is said to be *marked* if it has one or more tokens in a given marking.

Starting with some initial marking, the behavior of the Petri net evolves in a series of steps that alter the marking. The rule for changing the marking is simple: if all of the input places to a transition are marked, the transition is said to be *enabled*. An enabled transition can *fire*, which removes a token from each input place and adds a token to each output place. A marking of a very small Petri net is shown in Fig. 1(a). Figures 1(b) and 1(c) show the results of firing transitions t_0 and t_1 , respectively.

Any tokens not removed or added by firing a transition remain unchanged. Transitions fire one-at-a-time, so firing one transition may disable other transitions by removing tokens from their inputs, as well as enabling other transitions by adding tokens to their inputs. In Fig. 1, firing either transition disables the other.

Given a Petri net and an initial marking, a *firing sequence* is a list of transitions which can be fired in sequential order. There are usually many possible firing sequences, because whenever several transitions are enabled in the same marking, there is a choice of which transition to fire, and each such choice results in a different firing sequence. A Petri net marking m is said to be *reachable* from an initial marking if, from that initial marking, there is some firing sequence that results in m when the last transition fires.

A marking m that is the result of some firing sequence is said to be *reachable*. The set of reachable markings is generally much larger than the Petri net. For the 1-safe Petri nets used by the Pathalyzer, there can be as many as 2^n markings for n places. The problem of discovering whether a particular marking is reachable, or whether there is a reachable marking meeting a certain property, is computationally difficult – even the best known algorithms must search an exponential number of markings in the worst case.

There are many different types of Petri nets, and various applications often use Petri nets that are restricted in various ways. The Petri nets in the Pathalyzer are very simple and highly restricted. For example, unlike some Petri nets, they do not model quantitative time, the tokens are not labeled with data, and the transitions are not labeled with firing probabilities. These Petri nets are further limited since each place can only be marked with zero or one token. Petri nets with this property are called *safe* or *1-bounded* Petri nets. These restrictions enable analysis methods that would otherwise be difficult or impossible to implement efficiently.

2.2 Petri Nets for Biological Modeling

Petri nets are very general. The same system can be represented in many different ways using Petri nets, and different aspects of a system can be represented. Furthermore, the basic idea of a Petri net has been augmented in countless ways: adding time, probability, logical conditions on transitions, labeling tokens with data values, etc. There is a voluminous literature on Petri Nets [21].

There are many variants of the Petri net formalism and a variety of languages and tools for specification and analysis of systems using Petri nets. Petri nets have a graphical representation that corresponds naturally to conventional representations of biochemical networks. They have been used to model metabolic pathways and simple genetic networks [10,17,9,11,12,5,8,6]. These studies have been largely concerned with dynamic or kinetic models of biochemistry, so the models are qualitatively different from those used by the Pathalyzer. The questions answered by these previous efforts, and the types of analysis applied to the net, are completely different from those of the Pathalyzer.

In other work, a more abstract and qualitative view has also been taken, mapping biochemical concepts such as stoichiometry, flux modes, and conservation relations to well-known Petri net theory concepts [23]. Generalized Petri nets have also been used to model higher-level processes, including Malaria parasites invading host erythrocytes [16].

The Pathalyzer is unique in that its analysis depends on solving the *reachability problem* for Petri nets. The Pathalyzer can reliably answer questions about all possible pathways that can evolve from a defined starting state of a model and reach a cell state satisfying specified conditions. Pathways are discovered by solving the *Petri net reachability problem*. Since this problem is computationally difficult and the EGFR network has hundreds of reactions, it would not be unreasonable to guess that the problem could not be even solved in practice. Surprisingly, the pathalyzer is able to answer such queries in a few seconds, so a pathway can be displayed interactively (if such a pathway exists). This effect is achieved through the use of sophisticated heuristics developed in the Petri net community over several decades, which happen to work surprisingly well for the Pathalyzer's models.

2.3 Modeling Pathways

As mentioned above, pathways are not modeled explicitly in Pathway Logic or using the Pathalyzer. Instead, a collection of related reactions or processes is modeled: The

user queries the system as to whether something of interest can occur in response to a specific stimulus and, if the answer is affirmative, the Pathalyzer will display a pathway that achieves the result. This perspective has several advantages. First, it saves the redundant work entering individual pathways as data, including the problems of resolving inconsistencies between the pathways and reactions that comprise them. Second, because a large system of reactions/processes has a potentially huge number of similar pathways, representing them individually as data would be very inefficient. The Pathalyzer can dynamically generate pathways in response to user requests instead of storing them.

Here, the reactions/processes in a signaling network are transitions in a Petri net. Each place represents a combination of three factors: a molecule (e.g., the signaling protein *Gab1*), a possible modification (e.g. 'activated'), and a particular location within the cell (e.g., at the cell membrane). The transitions form a network because the places connect the reactions. For example, the product of one reaction is usually the substrate of another reaction.

In the Pathalyzer, the diagrams of Petri nets are drawn slightly differently from the examples of Fig. 1. The places are drawn as ellipses and labeled with the combination of molecule, location, and modification that they represent. The transitions are labeled with the reaction/process they represent. Any place that is both an input and output to a transition is depicted as an input to the transition with a dashed line; firing the transition will not unmark such a place (the conventional way to draw this situation would be to have arrows in both directions or a double-headed arrow, which tends to clutter the diagrams). For the analysis, some combinations of molecules and modifications in certain locations are assumed to be present initially. This situation determines the initial marking of the Petri net. Places that are initially marked are colored gray in the graphs generated by the tool. Figure 2 represents the reaction *353.Egfr.act.Gab1*, which is enabled when EGFR is activated in the cell membrane (*EGFRact(CM)*), *Gab1* is present in the cytoplasm (*Gab1(cyto)*), and *Grb2* is present in the cell membrane (*Grb2(CM)*). When the reaction fires, it activates *Gab1* and recruits it to the cell membrane (*Gab1act(CM)*). In the process, *Gab1(cyto)* is used up (there is a solid arrow from *Gab1(cyto)* to the reaction), but *EGFRact(CM)* and *Grb2(CM)* are not (they have dashed arrows). *Gab1(cyto)* is initially present (so it is gray), but *EGFRact(CM)* and *Grb2(CM)* are not.

Hundreds of reactions/processes have been curated, making up pathways involved in signaling from the activated EGFR. The tool displays all of these reactions as a graph.

3 Analysis Methods

A primary advantage of Petri nets is that they can be analyzed efficiently to answer non-trivial questions about the behavior of multiple interacting signaling pathways. Several kinds of analysis have been implemented in the Pathalyzer.

The tool is predicated on the assumption that the user is interested in discovering whether certain cell states can be reached, and, if so, how they can be reached. Such queries can reveal the possible interactions between pathways and show the effects of perturbations on pathways. In practice, the analysis methods have also been very helpful for checking the accuracy of the rules entered into the system.

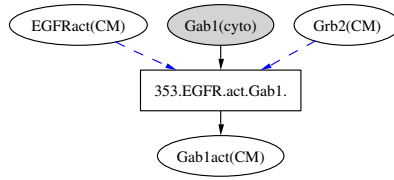


Fig. 2. A single reaction in a model, with associated substances. The inputs of the reaction are activated EGFR in the cell membrane, Gab1 in the cytoplasm, and Grb2 at the cell membrane. The reaction represents the recruitment of Gab1 to the cell membrane by Grb2 and the phosphorylation of Gab1 by EGFR. In the reaction, Gab1 is used up but the other two inputs are not.

All of the analysis methods are based on the user specifying one or more *goals*, which are places that are to be marked simultaneously in some reachable marking of the network, and *avoid places*, which are places that are not to be marked in the process of reaching a goal (*avoid places* will be treated as being unmarked initially, even if the Petri net includes them in the initial marking). For example, the user may be interested in whether it is possible for the transcription factors *cJun* and *cFos* to be activated in the nucleus at the same time.

3.1 Relevant Subnet

There are several queries that can be automatically answered given a particular goal. The simplest is: “What part of the net is relevant to achieving this goal?” The relevant subset of the net is often much smaller and easier to comprehend than the full net, which is usually quite large.

The determination of what is relevant is a two-stage process. First, places are classified as *potentially markable*. This set is approximate: every place that can be marked starting from the initial marking by some sequence of transition firings is guaranteed to be classified as potentially markable, but there may be potentially markable places that cannot actually be marked in a firing sequence. In practice, this approximation has turned out to be very accurate.

The computation of the potentially markable places is based on iterating two simple rules: a place is potentially markable if it is in the initial marking or if some input transition is potentially fireable. A transition is potentially fireable if *all* of its input places are potentially markable. The algorithm uses these rules to enlarge the set of potentially markable places and potentially fireable transitions until no more places or transitions can be added. If one of the goal places is not potentially markable, the goal cannot be achieved, and the relevant subnet is empty. This fact is reported to the user.

If all of the goal places are potentially markable, the second stage of the analysis is executed. This stage searches backwards from the goal places for places and transitions that are relevant, again using simple rules: all goal places are relevant and all input places of a relevant transition are relevant, and all input transitions in to a relevant place are relevant. Once again, the rules are applied repeatedly until no more places and transitions can be added to the relevant set.

3.2 Finding a Pathway

Another useful kind of analysis is to search for a particular pathway that leads to specified goals. This analysis first finds the relevant subset of the Petri net as described above, and then searches for a firing sequence starting from the initial marking that results in the goal places being marked.

Finding a pathway can be computationally difficult. The problem of whether a particular marking can be reached from a given initial marking is NP-complete even for safe acyclic Petri nets (and the nets in the Pathalyzer have cycles, which makes them harder to analyze). The obvious algorithm would be to search the graph of all reachable markings, saving the markings in a table to avoid redundant computation. Unfortunately, this method is slow and often does not complete because relatively small Petri nets can have large sets of reachable markings.

Fortunately, efficient algorithms have been devised over the years for solving reachability problems in Petri nets. The approach we use, with great success, is called *stubborn set reduction*. It searches the set of markings selectively, avoiding redundancy stemming from non-interacting transitions. Stubborn set reduction is subtle, so it is not described in more detail here; the interested reader can learn the details elsewhere [15]. In this application, stubborn set reduction works so well that there is little noticeable delay in finding a pathway to a particular goal, if it exists.

The firing sequence that reaches a specified goal would not be easy to interpret if it were just printed as a list. Instead, the tool supports visualization of the pathway by displaying the subset of the original graph consisting of those reactions appearing in the firing sequence and their associated input and output places. Unlike the firing sequence, which has many reactions/processes that occur in arbitrary order, the graphical display shows the dependencies between them, and shows when they are independent. This display is intended to depict a signaling pathway in a way familiar to most biological researchers. Fig. 3 shows a pathway in our EGFR model that leads to *cFos* being activated in the nucleus.

3.3 Avoid Places

One problem is that there are often many different pathways to the same goal. Users would like to have more control over which pathways are displayed by the system. In addition, users may wish to experiment *in silico* by examining the results of perturbations on the net. To provide this control, the Pathalyzer allows the user to specify additional places to *avoid*, by checking a box in the same menu as the goals. A place marked as “avoid” is not allowed to appear in the relevant subset of the net or in a pathway. If the goal cannot be achieved without using an avoid place, it is reported to be unreachable.

Avoid places are simple to implement in the relevant subset computation. The rules are modified so that avoid places are never added to the potentially markable set, and their input transitions are never added to the potentially fireable set (if any of these transitions were to fire, it would mark an avoid place). The effect of these changes is a list of additional places and transitions to be omitted from both phases of the relevant subnet computation.

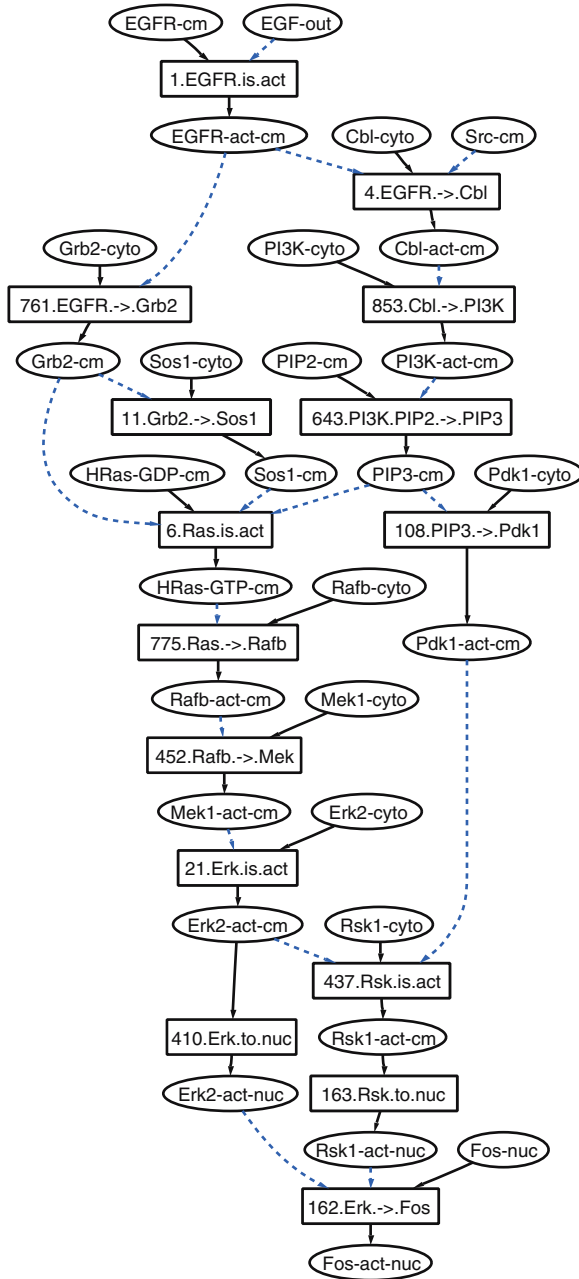


Fig. 3. A pathway as displayed by the tool that shows how the presence of EGF outside the cell can cause *cFos* to be activated in the cell nucleus

No additional effort is required to exclude avoid places in the search for a pathway, since the search is performed on the relevant subset of the net, which has had the avoid

places and their associated reactions removed before the pathway search. Hence, no avoid place will appear in a pathway displayed by this command.

Avoid places are useful to reduce clutter in the relevant net, search for a particular pathway, or explore redundancy in the pathways. The typical use of avoid places is to start with goals only and find the relevant subnet or a pathway. Unwanted places are identified in the result and designated as avoid places, then the command is re-run. Typically, it will display a smaller relevant net or a different pathway. Additional avoid places can be selected and the process repeated until the desired result is displayed, or the goals are found to be unreachable.

3.4 Knockout Search

Finally, there is one kind of analysis (so far) that is more automatic and global than searching for a pathway. The user may be interested in whether removing one or more signaling molecules renders a particular goal unreachable. For example, this may be interesting to predict the behavior of gene knockouts or identify drug targets. The Pathalyzer has a command to find all “single and double knockouts” relative to a particular specification of goal and avoid places.

The basic method for finding them is simply to automate the process by which it would be done manually. There is a loop that systematically selects one or two additional avoid places and attempts to find a pathway to the goal, logging all failures. The results are then listed for the user. This computation is greatly accelerated by observing that a single knockout *must* appear in every pathway to the goal, so the Pathalyzer searches for a pathway first, then only considers as knockouts places that appear in that pathway.

4 Implementation

The tool is implemented primarily in Java, although it uses two other programs. The structure of the Petri net and its initial marking is stored in an XML file, which is read by the tool. All graphs are laid out by the **dot** program from AT&T Bell Laboratories (it is available free as part of the GraphViz suite of tools from AT&T Bell Labs [22]). The Java interface provides an interactive display for the graphs using the Java-2D graphics library.

Because the graph of an entire network can be too large to fit conveniently on a display screen, the Pathalyzer makes it possible to zoom in and out, and to select a node that is to remain centered at all times. There is also a menu to search for a node by name; if the node is found, it is highlighted in the network. The nodes immediately connected to a particular node can be displayed, and then the user can interactively enlarge the fragment of the graph that is displayed.

The user selects the goal and avoid places from a menu that lists all of the places, and provides buttons to selection them as “goal,” “avoid,” or “neither.” Once goals and avoids are selected, they persist until they are changed.

The user can cause the relevant subnet to be computed relative to the currently selected goal and avoid places, and then displayed in a separate window. The user can also

request that a pathway be found that reaches the specified goals while not marking any of the avoid places. If the goals are in the relevant subnet, the tool invokes LoLA, a Petri net analysis tool that does stubborn set reduction. LoLA is available under GNU General Public License license [20]. If LoLA is able to find a firing sequence to the goals, it stores it in a temporary file which is read by the Java interface. The Java program then finds the subnet of the full Petri net corresponding to the firing sequence, prunes any unnecessary transitions by re-running the relevant subnet algorithm on it, and displays it in a new window using the same code as the display of the original network. For convenience, new analysis commands can be processed from the window displaying the pathway, working from the goals and avoids that resulted in that pathway.

Users often want to compare the pathways that are found by the tool with different choices of avoid places. The tool allows the two most recently computed pathways to be compared. The comparison display shows the nodes from both graphs, and color-codes those that appear only in the first graph, only in the second graph, or in both graphs.

The user can also request the single and double knockouts that would prevent the currently selected goals from being achieved given the currently selected avoid places. The program pops up a window with a simple textual list of the individual places and pairs of places that prevent the goal from occurring.

5 Discussion

At this point, the Pathalyzer is a demonstration prototype. The tool could be used in a variety of ways. At the very least, it could be valuable an interactive “desk reference,” summarizing the vast literature on signaling pathways. If so, an important feature would be to link the reactions to the relevant source literature.

More ambitiously, the Pathalyzer could be used by researchers to interpret the results of experiments, and generate hypotheses for new experiments. For example, various reactions could be knocked out, and the results predicted by the program could be compared with results in the laboratory to test the Pathalyzer’s model. Inconsistencies with experiments indicate a need to refine the model (or the experiments).

Answering the above question requires developing more comprehensive models, which will require expanding the community of curators. There needs to be a system for entering new models that is accessible to biologists who are not experts in Maude. Such a system would also allow researchers to develop pathway models focusing on their particular areas of interest. Work is underway on this question.

It is easy to imagine extensions for modeling reaction rates, quantitative time, and so on. However, any such additions makes analysis more difficult. The longer-run research challenge is to strike the correct balance between modeling power and analytical power, which can only be found with more use.

Acknowledgments

Some of the early ideas about using Petri nets for this application emerged in discussions with Chris Myers in 2002. Much of the research and programming was done while the first author was on sabbatical at SRI, International in 2003.

Work by the first author on this publication was partially supported by NIH Grant Number NIH 5 U56 CA 112973 from the ICBP Program. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of NIH.

References

1. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. Maude 2.0 Manual, 2003. <http://maude.cs.uiuc.edu>.
2. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott. The Maude 2.0 system. In *Rewriting Techniques and Applications (RTA'03)*, Lecture Notes in Computer Science. Springer-Verlag, 2003.
3. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, José Meseguer, and Kemal Sonmez. Pathway Logic: Symbolic analysis of biological signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, January 2002.
4. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, and Carolyn Talcott. Pathway Logic: Executable models of biological networks. In *Fourth International Workshop on Rewriting Logic and Its Applications (WRLA'2002)*, Pisa, Italy, September 19 — 21, 2002, volume 71 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002. <http://www.elsevier.nl/locate/entcs/volume71.html>.
5. J. S. Oliveira et al. An algebraic-combinatorial model for the identification and mapping of biochemical pathways. *Bull. Math. Biol.*, 63:1163–1196, 2001.
6. J. S. Oliveira et al. A computational model for the identification of biochemical pathways in the Krebs cycle. *J. Computational Biology*, 10:57–82, 2003.
7. M. Hucka et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *bioinformatics*, 19(4):524–531, 2003.
8. H. Genrich, R. Küffner, and K. Voss. Executable Petri net models for the analysis of metabolic pathways. *Int. J. STTT*, 3, 2001.
9. P. J. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology using stochastic Petri nets. *Proc. Natl. Acad. Sci. U. S. A.*, 95:6750–6755, 1998.
10. R Hofestädt. A Petri net application to model metabolic processes. *Syst. Anal. Mod. Simul.*, 16:113–122, 1994.
11. R. Küffner, R. Zimmer, and T. Lengauer. Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics*, 16:825–836, 2000.
12. H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid Petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing*, volume 5, pages 341–352, 2000.
13. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.
14. José Meseguer. Rewriting logic and Maude: A wide-spectrum semantic framework for object-based distributed systems. In S. Smith and C.L. Talcott, editors, *Formal Methods for Open Object-based Distributed Systems, FMOODS 2000*, pages 89–117. Kluwer, 2000.
15. Mogens Nielsen and Dan Simpson, editors. *LoLA: A Low Level Analyser*, volume 1825 of *Lecture Notes in Computer Science*, 2000.
16. Mor Peleg, Iwei Yeh, and Russ B. Altman. Modelling biological processes using workflow and Petri Net models. *Bioinformatics*, 18(6):852–837, 2002.
17. V. N. Reddy, M. N. Liebmann, and M. L. Mavrouniotis. Qualitative analysis of biochemical reaction systems. *Comput. Biol. Med.*, 26:9–24, 1996.
18. C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Proceedings of the Pacific Symposium on Biocomputing*, January 2004. to appear.

19. <http://www.csl.sri.com/users/clk/PLweb/pl.html>, Pathway logic.
20. <http://www.informatik.hu-berlin.de/top/lola/lola.html>, Lola system.
21. <http://www.informatik.uni-hamburg.de/TGI/pnbib/>, The Petri Nets Bibliography.
22. <http://www.research.att.com/sw/tools/graphviz/>, The GraphViz Homepage.
23. I. Zevedei-Oancea and S. Schuster. Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology*, 3(0029), 2003.

Decomposition of Overlapping Protein Complexes: A Graph Theoretical Method for Analyzing Static and Dynamic Protein Associations

Elena Zotenko^{1,2}, Katia S. Guimarães^{1,3},
Raja Jothi¹, and Teresa M. Przytycka¹

¹ NCBI/NLM/NIH, Bethesda, USA

² Department of Computer Science, University of Maryland, College Park, USA

³ Center of Informatics, Federal University of Pernambuco, Recife, Brazil

Abstract. We propose a new method for identifying and representing overlapping *functional groups* within a protein interaction network. We develop a graph-theoretical framework that enables automatic construction of such representation. The proposed representation helps in understanding the transitions between functional groups and allows for tracking a protein's path through a cascade of functional groups. Therefore, depending on the nature of the network, our representation is capable of elucidating temporal relations between functional groups. We illustrate the effectiveness of our method by applying it to TNF α /NF- κ B and pheromone signaling pathways.

1 Introduction

A major challenge in systems biology is to understand the intricate network of interacting molecules. The complexity in biological systems arises not only from various individual protein molecules but also from their organization into systems with numerous interacting partners. In fact, most cellular processes are carried out by multi-protein complexes, groups of proteins that bind together to perform a specific task. Some proteins form stable complexes, such as the ribosomal complex that consists of more than 50 proteins and three RNA molecules, while other proteins form transient associations and are part of several complexes at different stages of a cellular process. A better understanding of this higher-order organization of proteins into overlapping complexes is an important step towards unveiling functional and evolutionary mechanisms behind biological networks.

Data on protein complexes are collected from individual systems study, and more recently through high-throughput experiments, such as yeast two-hybrid (Y2H) [23, 15] and tandem affinity purification followed by mass spectrometry (TAP/MS) [13, 10]. The TAP/MS approach helps pinpoint proteins that interact with a tagged *bait* protein, either directly or indirectly, and are thus suited to identify multi-protein complexes. In fact, several research groups have systematically applied TAP/MS technology to study protein complexes involved in different signaling pathways [4].

Protein interactions are routinely represented as graphs, with proteins as nodes and interactions as edges (links). Therefore, it is not surprising that analysis of protein interaction networks reach out for a variety of graph-theoretical tools. Following the observation that protein interaction networks display a characteristic power-law like node degree distribution [2], a substantial body of research focused on statistical properties of protein interaction networks [17, 18]. Subsequently, several computational methods to identify network modules and/or protein complexes have been developed. In a protein interaction network, such modules correspond to densely connected subgraphs, either *cliques* or “*cliquish*” components. This observation serves as a starting point for several computational methods to identify protein complexes and families of overlapping protein complexes in high-throughput protein interaction networks [1, 19, 5, 21]. Such cliques or densely connected subgraphs may contain, in addition to proteins that form a molecular complex, proteins that transiently interact with other proteins. Therefore, Spirin *et al.* [21] use the term *functional module* to denote groups of proteins which are densely connected within themselves but sparsely connected with the rest of the network. Alternatively, Tornow *et al.* [22] defines a functional module as a group of genes or their products in a metabolic or signaling pathway, which are related by one or more genetic or cellular interactions, e.g. co-regulation, co-expression or membership in a protein complex, and whose members have more relations among themselves than with members of other modules. While the definition of a functional module is neither formal nor precise, it is generally uncontroversial.

In this work, we model a functional module as a union of overlapping dense subnetworks called here *functional groups*. A functional group is either a maximal clique (typically representing a protein complex) or a set of alternative variants of such complexes/cliques. As components of a larger functional module, functional groups are not assumed to be well separated and can have significant overlaps. Intuitively, if a functional module performs a function that requires a sequence of steps (like in the case of a signaling pathway) then we would like functional groups to be snapshots of protein associations at these steps.

In a recent paper, Gagneur *et al.* applied *modular decomposition* to elucidate the organization of protein complexes [9]. The basic principle behind modular decomposition is to iteratively identify and contract nodes that are equivalent in certain sense, until no more equivalent nodes can be found in the graph. A graph is called *prime* if it cannot be decomposed any further. Graphs that belong to a very special family called *cographs* are the only ones that can be completely decomposed (that is, the iterative reduction process does not halt on a non-trivial prime graph). While modular decomposition provides an excellent description of combinatorial variants within a family of complexes, it cannot help in elucidating the manner in which proteins participate in dynamically changing complexes, which is particularly interesting if the family of complexes represents a temporal relation of various stages in a dynamically changing complex.

We propose a new method for identifying and representing overlapping functional groups in a functional module. Furthermore, if the module corresponds

to a dynamic process that requires certain complexes (or more generally functional groups) come into contact in a specific order, our method attempts to discover this order. Our method is motivated by a fundamental result for *chordal graphs* [11], which states that every chordal graph has the so called *clique tree* representation. However, not every protein interaction network is chordal and not every functional group is a clique. Therefore, we developed a graph-theoretical framework that enables automatic construction of tree-like representation, analogous to the clique tree representation, for much broader family of graphs. We call this tree representation *Tree of Complexes*. The nodes in the tree are functional groups, and for every protein, the set of functional groups that contain this protein forms a single subtree. The “single subtree” requirement restricts significantly the way in which the nodes of the tree can be interconnected. As a consequence, this representation shows a smooth transition between functional groups and allows for tracking a protein’s path through a cascade of functional groups. Moreover, depending on the nature of the network, the representation may be capable of elucidating temporal relations between functional groups.

We developed a new method, *Complex Overlap Decomposition* (COD), that given a protein interaction network identifies its functional groups and constructs the Tree of Complexes representation. Our method requires that the network satisfies certain mathematical properties. Mathematical properties of this graph family and the COD method are discussed in the Results section.

We applied the COD method to several protein interaction networks, such as the TNF α /NF- κ B signaling pathway and the pheromone signaling pathway. The corresponding subnetworks for all interaction networks are extracted from high throughput experimental data. Our results show that the COD method opens a new avenue for the analysis of protein interaction networks.

2 Results

One way to represent a set of overlapping functional groups is to construct a graph with nodes representing functional groups and edges representing overlaps, i.e., there exists an edge between two functional groups if and only if they share at least one protein. This approach has two shortcomings. First, it is not obvious how to correctly identify functional groups, and second, such a representation does not provide any information about the dynamics of proteins in the network. We propose a graph-theoretical approach, which, under the assumption that the protein interaction network satisfies certain mathematical properties, identifies functional groups and provides a representation of overlaps between functional groups in the form of the Tree of Complexes.

Here, we first describe the COD method. Then, we demonstrate the utility of our approach by applying the COD method to several examples, derived from high-throughput experiments, TNF α /NF- κ B and pheromone signaling pathway interaction networks.

2.1 Complex Overlap Decomposition

Our method of representing overlapping functional groups, which is depicted in Figure 1, builds on chordal and cograph graph theories. Chordal graphs constitute an important and well studied graph family [12, 20]. A *chord* in a graph is any edge that connects two non-consecutive nodes of a cycle. A *chordal graph* is a graph which does not contain chordless cycles of length greater than three.

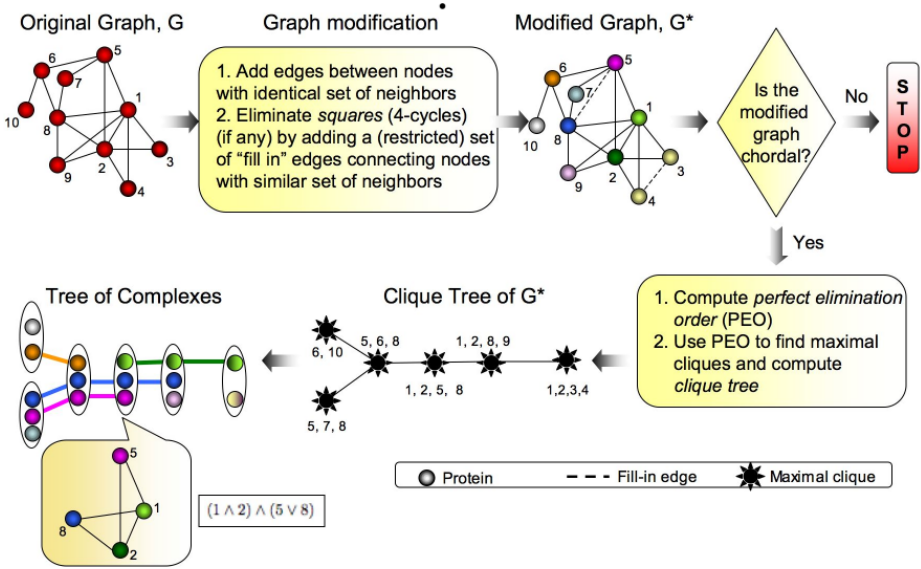


Fig. 1. A simplified illustration of the Complex Overlap Decomposition (COD) method. An edge, (3,4), connecting a pair of weak siblings is added to the graph. A fill-in edge between proteins 5 and 8 is added to eliminate all five 4-cycles in the graph: {5, 6, 8, 7}, {1, 5, 7, 8}, {2, 5, 7, 8}, {1, 5, 6, 8}, and {2, 5, 6, 8}. If the modified graph is chordal, perfect elimination order is used to identify all maximal cliques and construct a clique tree. The Tree of Complexes is constructed by projecting each maximal clique in the modified graph, G^* , to a functional group in the original graph G . For example, a four node maximal clique, {1, 2, 5, 8}, in G^* is projected to a four node functional group in G , by removing a fill-in edge (5, 8). Each functional group is represented by a Boolean expression, such as $(1 \wedge 2) \wedge (5 \vee 8)$, which means that the functional group contains two variants of a complex, {1, 2, 5} and {1, 2, 8}.

An important property of chordal graphs, which is explored directly in this paper, is that every chordal graph has a corresponding *clique tree* [11]. The nodes in the tree are maximal cliques. Moreover, for every node in the graph, a set of maximal cliques that contain this node form a connected subgraph of the clique tree. Thus, there is a mapping between the nodes in the graph and subtrees in the clique tree. The “connected subgraph” requirement puts constraints on the topology of the clique tree. In fact, the topology of the tree is determined by the structure of overlaps between the maximal cliques in the graph. Thus, the clique

tree captures information about the structure of the overlaps, which is lost in a simple clique intersection graph as shown in the example below.

Example. Consider a hypothetical protein interaction network in Figure 2(a). This network is chordal and its maximal cliques are listed in Figure 2(b). We want to contrast the clique tree representation in Figure 2(d) to a naive representation in Figure 2(c), where every pair of maximal cliques that contain a protein in common is connected by an edge.

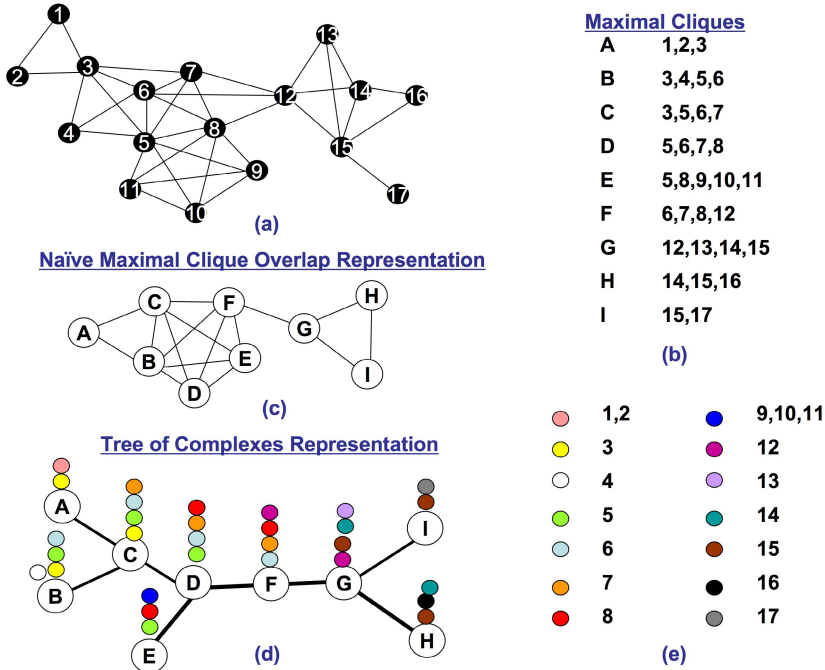


Fig. 2. (a) A hypothetical protein interaction network. (b) A list of all maximal cliques in the network. (c) A naive representation of overlaps between maximal cliques. Each maximal clique is a node and there is an edge between two maximal cliques if and only if they share a protein. (d) The clique tree representation. Once again, every maximal clique is a node, but the cliques are connected in such a way that the resulting graph is a tree. Moreover, cliques that contain a given protein form a connected subgraph. (e) This color scheme is used to show the subtree of every protein. For example, protein 3 is contained in maximal cliques A, B, and C, which is shown by placing yellow dots above the maximal cliques.

While both representations show the overlap between maximal cliques, the interconnection pattern of cliques in the naive representation carries little additional information about the structure of this overlap. On the other hand, a very specific tree-like interconnection pattern in the clique tree representation can expose a special structure of such overlap. For example, consider maximal

cliques B through F . In the naive representation, the overlap between these maximal cliques is collapsed to a clique. Thus, the representation treats the maximal cliques and overlaps between them equally. In particular, there is no way to tell that, for example, D occupies a more central position in the network than B . In the clique tree representation this information can be extracted from the relative position of cliques in the tree. For example, B is connected to F by a path that passes through C and D , which means that any protein shared by B and F is also contained in C and D . In other words, the overlap between B and F is entirely contained in the overlap between B and D , which in turn is entirely contained in the overlap between B and C . Thus, there is a correlation between the amount of overlap between maximal cliques and their distance in the clique tree.

Nice properties of the clique tree mentioned above make it a good choice for representation of overlaps between functional groups. However, not every protein interaction network is chordal and maximal cliques may not always be the best way to represent functional groups. For example, in Figure II, cliques 1,2,3 and 1,2,4 may correspond to two variants of one complex, where proteins 3 and 4 replace each other. Therefore, in the COD decomposition we relax the assumption that the functional groups are maximal cliques, and allow them to have some variability. To capture systematically variants within a functional group we represent it with a graph from a family of graphs known as cographs.

Cographs are another well-studied graph family [6]. A *cograph* can be characterized by an absence of induced path of length four (P_4), where length is measured by the number of nodes in the path. Thus, the diameter of a connected cograph is at most two. Subsequently, connected cographs are dense and cliquish, consistently with the assumption made by algorithms that delineate protein complexes. What makes cographs even more attractive is that for every cograph there exists a Boolean expression which describes the maximal cliques in the graph. (In terms of modular decomposition used in [9] it means that a cograph can be decomposed by modular decomposition without leaving non-trivial non-decomposable prime module.) This Boolean expression describes in a compact and hierarchical way all possible variants of a functional group.

The main idea behind COD method is to provide a representation of a functional module, which is analogous to the clique tree, but in which nodes are cographs (representing variants of a functional group) rather than maximal cliques. If we knew in advance all functional groups in the module, then we could simply connect the proteins within each functional group turning it into a clique and, under the assumption that the resulting graph is chordal, apply clique tree construction algorithm to the graph. Since we do not have predefined functional groups, our algorithm identifies them by adding edges to the graph in such a way that each added edge connects a pair of nodes that putatively belong to the same functional group.

The COD method's edge addition strategy and its biological motivation builds on a concept of *weak siblings*. We call a pair of nodes weak siblings if and only if they are connected to the exactly the same set of neighbors, but are not

connected to each other. In terms of protein interaction networks, weak siblings are proteins which interact with the same set of proteins but do not interact with each other. In particular, proteins that can substitute each other in a protein interaction network may have this property. Similarly, a pair of proteins that belong to the same complex but are not connected due to missing data or an experimental error will be represented as weak siblings. Since the weak siblings relationship suggests functional similarity, the COD method takes a first step towards delineation of functional groups by connecting every pair of weak siblings by an edge. As this modification may also eliminate some of the squares in the graph, functional group delineation happens simultaneously with transformation of the protein interaction graph into a chordal graph.

If, after connecting all pairs of weak siblings, the resulting graph is not chordal, the COD method attempts to transform it to chordal by adding some additional edges. Consistently with our assumption that we connect only nodes corresponding to proteins that could be put in the same functional group, we impose restrictions on this “fill-in” process. Namely, we require that, each introduced edge connects a pair of nodes which is close to being weak siblings. In such case the new edge is a diagonal of one or more squares in the graph. We emphasize that adding edges between nodes of longer cycles has no such justification.

To summarize our edge addition procedure, our method attempts to eliminate all the squares in the protein interaction network by adding a limited set of diagonals that satisfies following conditions (i) connects potentially functionally equivalent proteins, as measured by the overlap in neighborhoods or distance from being a pair of weak siblings; (ii) ensures that functional groups correspond to cographs; we argue that this condition is guaranteed if the set of added edges does not form a P_4 in a maximal clique of the modified graph (cf. Materials and Methods).

If the modification step succeeds, i.e., the modified graph is chordal, the clique tree representation of the modified graph is constructed and then extended to the Tree of Complexes representation of the original graph. The COD algorithm keeps track of all the edge additions and uses this information to delineate functional groups by projecting each maximal clique onto original network and removing all introduced edges contained in the clique. For example, in the modified graph of Figure 1 a maximal clique with four nodes, $\{1, 2, 5, 8\}$, is projected to a functional group by removing an edge connecting protein 5 and 8. This functional group contains two variants of a protein complex, $\{1, 2, 5\}$ and $\{1, 2, 8\}$, which are compactly represented by a $(1 \wedge 2) \wedge (5 \vee 8)$ Boolean expression. If, on the other hand, the modified graph is not chordal, the COD method stops without producing the representation.

Since the clique tree representation for a chordal graph is not unique, the Tree of Complexes representation that derives from it is not unique either. The clique tree topology is determined by the “connected subgraph” constraints and restriction power of these constraints depends on the structure of the underlying graph, i.e., there are graphs with a unique tree topology and there are graphs for which almost any tree that spans all the maximal cliques in the graph is a

valid clique tree. For every protein interaction network below we explicitly state all the possible Tree of Complexes representations.

2.2 Applying COD to Protein Interaction Networks

TNF α /NF- κ B Signaling Pathway. To illustrate the power of COD in elucidating the dynamics behind protein complexes, we consider the TNF α /NF- κ B signaling pathway. The Nuclear Factor κ B (NF- κ B) family of transcription factors is activated in response to a diverse set of stress stimuli, which includes pro-inflammatory cytokines, *e.g.*, TNF α . In vertebrates, this family includes p50, p52, Rel A, c-Rel, and Rel B, which bind to the DNA in a homo or heterodimeric fashion. NF- κ B activity is regulated by the I κ B family of proteins via inhibitory ankyrin repeat domains. This family includes I κ B α , I κ B β , and I κ B ϵ . The precursors of p50 (p105) and p52 (p100) also possess ankyrin repeat domains and thus act as inhibitors. These precursors can also form dimers with other members of the NF- κ B family. The activation with the pro-inflammatory cytokine tumor necrosis factor TNF α triggers a signaling cascade, which, in particular, stimulates the activation of the IKK α , IKK β , and IKK γ functional groups. The IKKs initiate a signal induced degradation of the inhibitors (I κ Bs), and subsequent nuclear translocation of the transcription factor. Recent TAP experiments [4] provide a wealth of new information regarding this important signaling pathway. Bouwmeester *et al.* identified 221 molecular associations, out of which only 80 were previously known. Gagneur *et al.* [9] applied modular decomposition to the network of these associations but the decomposition halted quickly at large non-decomposable modules.

We used the COD method to analyze the subnetwork spanning all the paths from NIK (NF κ B-inducing kinase phosphorylating IKK α and IKK β) with at most three edges. For the purpose of the analysis, we contracted all five members of NF- κ B family into one node. As the resulting protein interaction network, shown in Figure 3(a), is chordal without weak siblings, functional groups correspond to maximal cliques in the network.

For this network, there are two alternative Tree of Complexes representations: functional group E can be connected to either D or C . The representation that maximizes the number of leaves is shown in Figure 3(b). One can clearly see the interplay between the activators and inhibitors. Proteins p105 and NF- κ B participate in the same functional groups and thus follow the same path in the tree. The same is true for the pair of proteins I κ B α and I κ B β . The Tree of Complexes captures this by grouping p105 and NF- κ B, and I κ B α and I κ B β .

Pheromone Signaling Pathway. The yeast *Saccharomyces cerevisiae* may be present in one of two haploid cell types, which are able to mate. Pheromones released by one type of cell bind to a specific receptor of the other type. This triggers the activation of a scaffold protein-bound mitogen-activated protein kinase (MAPK) cascade and subsequent activation of nuclear proteins that control subsequent cellular events. In a recent paper, Spirin *et al.* [21] identified a subnetwork of proteins involved in this process within a yeast protein interaction

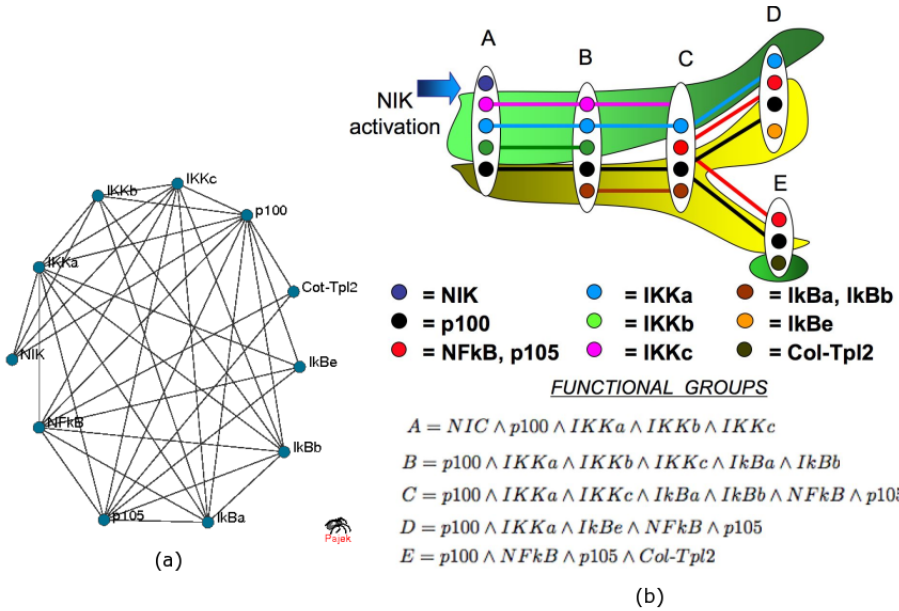


Fig. 3. A subnetwork of $TNF\alpha/NF-\kappa B$ signaling pathway. (a) The network. (b) The Tree of Complexes representation. The flow of action is visually represented by background colors: green for activators (IKKs) and yellow for inhibitors ($I\kappa B$ s, and p100). The NIK kinase is in the first functional group (A), together with all three members of the IKK complex and p100. Functional group B includes, in addition to p100, the IKKs and two inhibitors $I\kappa B\alpha$ and $I\kappa B\beta$. This group is the beginning of interaction between IKKs and $I\kappa B$ s. Functional group C loses some of the IKKs, continues to show $I\kappa B$ and begins to show interaction between $I\kappa B$ s and NF- κB factors. Finally, in group E we see the entrance of NIK-independent Col-Tpl2 kinase.

network [16]. We analyzed this subnetwork using the COD to see if our method can extract elements of temporal ordering. The subnetwork identified by Spirin *et al.* and its Tree of Complexes representation is given in Figure 4. In this case, the protein network is not chordal. First, the COD method identifies and connects a pair of weak siblings, $MKK1$ and $MKK2$. Then, to transform the network to a chordal graph, three additional edges are added: $(SPH1, SPA2)$, $(FUS3, KSS1)$, and $(STE11, STE7)$. In this case, some functional groups will contain more than one protein complex and for each functional its Boolean expression is given. This network admits six different Tree of Complexes representations: (i) functional group H can be connected to either B or C ; (ii) any interconnection pattern that spans groups E , F , and G can be chosen. If we ask for a tree with maximum number of leaves, the number of tree variants is reduced to two (option (i)).

The MAPK cascade module consists of three sequentially acting protein kinases: MAP kinase kinase kinase (STE11) MAP kinase kinase (STE7) and MAP kinase (KSS1, FUS3) [24]. $MKK1$ and $MKK2$ are two redundant protein kinase

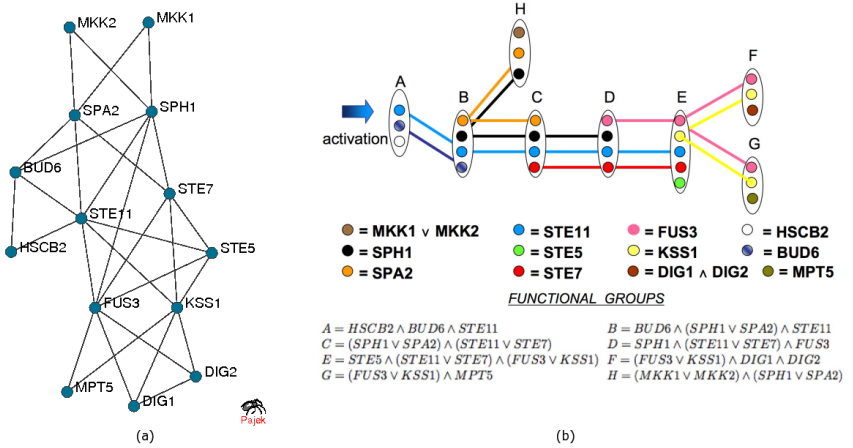


Fig. 4. Pheromone signaling pathway based on a subnetwork constructed using the results from high throughput experiments. (a) The network. (b) The Tree of Complexes representation. For the description of the elements of the tree see the text.

kinases (most similar to STE7) [14]. Their redundancy is properly captured by the \vee (*OR*) in their functional group (H). The MAP kinases KSS1 and FUS3 are two separate kinases both activated by STE7 each of which is essential for a different program: FUS3 - for mating; KSS1 - for the filamentous growth [8]. Once again this is correctly captured by \vee (*OR*) in groups F and G. STE5 is a scaffold protein of the MAPK module. It recruits MAPK module kinases (STE11, STE7, FUS3). This is consistent with the central position of the functional groups containing STE5 in the tree and relative position of paths of STE7, STE11 and FUS3 with respect to the path of STE5. Finally, nuclear proteins DIG1 and DIG2 (necessary for transcription inhibition, which are regulated by both FUS3 and KSS1) enter at the endpoints (nodes *F* and *G*) in the tree.

3 Materials and Methods

3.1 Computing Clique Tree

We use an elegant and efficient strategy for chordal graph recognition and clique tree construction outlined in [20].

3.2 A Compact Boolean Representation of Functional Groups

Recall that a functional group corresponds to a maximal clique in the modified protein interaction network, with modifications being addition of edges between every pair of weak siblings and then addition of edges that eliminate all the squares in the graph. The edge addition is such that no maximal clique in the modified graph contains an induced P_4 formed entirely by the added edges.

Following lemma guarantees that every functional group is a cograph and therefore admits a compact Boolean representation.

Lemma. For every functional group, a subgraph of the original graph induced by the members of the group contains an induced P_4 if and only if the set of edges added by our algorithm contains an induced P_4 .

Proof. The argument follows from Figure 5. Indeed, (v_1, v_2, v_3, v_4) is a P_4 in the original graph if and only if (v_3, v_1, v_4, v_2) is a P_4 formed by the added edges.

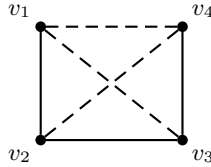


Fig. 5. A P_4 in the subgraph induced by the members of a functional group corresponds to a P_4 in the set of added edges. Solid lines correspond to the original edges and dashed lines correspond to the added edges.

3.3 Edge Addition

We use a reduction to the Minimum Vertex Cover problem to find all minimal sets of up to k edges that eliminate all the squares in the graph.

Each eliminating set of edges, $S = \{e_1, \dots, e_r\}$, is assigned a cost:

$$\text{cost}(S) = \sum_i (1.0 - \text{sim}(e_i)) ,$$

where $\text{sim}(e_i)$ takes values between 1.0 and 0.0, and measures our confidence in adding e_i to the graph. Since the addition of $e_i = (u_i, v_i)$ implies an interaction or functional equivalence between proteins u_i and v_i , we chose $\text{sim}(e_i)$ to be the amount of overlap between the neighborhoods of u_i and v_i , i.e., $\text{sim}(e_i) = \frac{|\mathcal{N}(u_i) \cap \mathcal{N}(v_i)|}{|\mathcal{N}(u_i) \cup \mathcal{N}(v_i)|}$, where $\mathcal{N}(v_i)$ denotes a set of neighbors of node v_i in the graph. Intuitively, $\text{sim}(e_i)$ measures how close u_i and v_i are to being a pair of weak siblings. If u_i and v_i have the same neighborhoods then $\text{sim}(e_i) = 1.0$; as the overlap between the neighborhoods decreases, $\text{sim}(e_i)$ goes to 0.0.

Then, we pick an edge set with the minimum cost from all the edge sets that do not form a P_4 , which is entirely contained in one of the maximal cliques of the modified graph. The last requirement is necessary to ensure that each functional group can be represented by a compact Boolean expression.

Reduction to the Minimum Vertex Cover. A square in a graph, a chordless cycle of length four, can be eliminated by adding one or both of its diagonals to the graph. For example, a graph in Figure 6 has two squares: (A, B, C, D) and (A, B, E, D) . Note that (B, C, D, E) is not a square as one of its diagonals, (C, E) , is an edge in the graph. The square (A, B, C, D) can be eliminated if

either edge (A, C) or (B, D) is added to the graph. Furthermore, a single diagonal can eliminate more than one square. For example, the diagonal (B, D) eliminates both squares. We are interested in finding all minimal sets of diagonals of size up to k that eliminate all the squares in the graph.

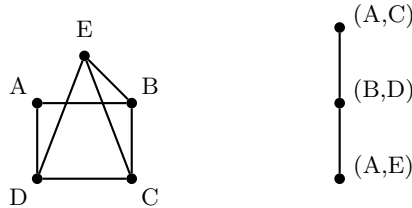


Fig. 6. A graph and a corresponding “square coverage graph”

We reduce the above problem to the Minimum Vertex Cover problem. The squares in the original graph become edges and diagonals become vertices in the new graph. Thus the original graph is transformed to a “square coverage” graph, which in turn serves as an input to the Minimum Vertex Cover problem. In the Minimum Vertex Cover problem we are given a graph and are asked to find the smallest set of vertices that cover all the edges in the graph. An edge is covered if at least one of its end points is selected. Coming back to our example, it can be easily seen that $\{(B, D)\}$ is the minimum vertex cover (Figure 6).

Although the Minimum Vertex Cover problem is an NP-hard problem, if the size of the optimum solution is small an efficient algorithm can be obtained. In other words the Minimum Vertex Cover problem is fixed parameter tractable. We use an $O(2^k(n + m))$ algorithm [7] to identify all minimal sets of edges of size up to k that eliminate all the squares in the graph.

3.4 Graphs That Have Tree of Complexes Representation

The COD method is not guaranteed to produce the Tree of Complexes representation for every possible protein interaction network. How can a family of graphs that admit Tree of Complexes representation be characterized? First, we argue that chordal graphs belong to this family. It can be shown that addition of edges that connect weak siblings does not introduce chordless cycles to the graph. Therefore, after all weak siblings are connected the graph is still chordal and thus admits Tree of Complexes representation. Next, we argue that cographs admit Tree of Complexes representation. Our methods eliminates all the squares in the graph, unless every possible set of eliminating edges forms a P_4 , which is entirely contained in one of the maximal cliques of the modified graph. It can be shown that the latter case is possible only when the original graph contains a P_4 , and thus is not a cograph. We conjecture that graphs that admit Tree of Complexes representation are exactly those graphs that admit a clique tree representation, with the nodes being maximal cographs rather than maximal cliques.

4 Discussion

Recent advances in experimental techniques resulted in the accumulation of a vast amount of protein interaction information, which is routinely represented by protein interaction networks. Therefore it is not surprising that increasingly more complex graph-theoretical tools are deployed to analyze protein interaction graphs and extract biologically meaningful patterns.

In general, graphs are not required to have any type of regularity. This makes them a very flexible tool which is able to represent complex relationships. However, this often also makes them computationally hard to deal with, for many problems in graph theory are NP-complete. Frequently graph theoretical problems can be simplified if some restrictions are imposed on the graph. Various restrictions give rise to various graph families. Given a graph family, it is usually very useful to be able to represent it using some kind of a tree. Such tree representation exposes a hierarchical organization that a graph may have, allowing for a structured analysis of it.

In this work we proposed a tree representation for protein interaction graphs called Tree of Complexes representation. Nodes in the Tree of Complexes are functional groups and the tree satisfies the additional condition that functional groups that contain any fixed protein form a connected subgraph. In this way, our representation captures not only the overlap between functional groups but, potentially, also the manner in which proteins enter and leave their enclosing functional groups. We developed a method (together with the corresponding graph-theoretical theory) for efficient identification of such overlapping functional groups and construction of the corresponding Tree of Complexes. In particular, our method differs from other approaches in that it does not attempt to enumerate disjoint complexes but instead identifies and represents relations between overlapping functional groups. Even though the Tree of Complexes representation is not unique, the protein interaction networks that we analyzed admit very few alternative tree topologies. If we ask for a tree topology with a maximum number of leaves, as not to impose an artificial order between functional groups, the number of tree topologies is reduced even further. Thus, in the TNF α /NF- κ B signalling pathway this results in a unique Tree of Complexes representation and in the pheromone signalling pathway in two very closely related possible Tree of Complexes representations.

The nature of high-throughput protein interaction data does not directly imply that this data encodes temporal relations. We demonstrated that our method is frequently capable of discovering such temporal relations. Interestingly, temporal associations can also be implicated in the absence of actual interaction in the data. For example, in the case of the pheromone signaling pathway, our method correctly included KSS1 and FUS3 in the same functional group (treated here as temporal associations), despite the fact that there is no link between KSS1 and FUS3 in the input protein interaction network.

Obviously, there are limitations to deciphering such temporal relations. For example, we cannot provide temporal ordering between different tree branches.

Furthermore, if a functional group is not a clique but is represented as a Boolean expression indicating various possibilities for such group, then one can not be sure if these variants are mutually exclusive or if they represent partial information capturing incomplete data. Even in the case when the functional unit forms a clique it is still possible that it contains interactions that are not simultaneous. For example, interactions between pairs (A,B), (B,C) and (C,A) are represent as a three-vertex clique with nodes A,B,C and thus cannot be distinguished from a trimer (A,B,C). Such coincidences are less likely for larger cliques.

Although our algorithm is not guaranteed to produce the Tree of Complexes representation for every possible protein interaction network, the algorithm will succeed for a broad family of graphs, which includes chordal graphs (and thus interval graphs) and cographs. Currently, our method can be applied to protein interaction networks that do not contain long (longer than four node) chordless cycles. As a consequence, it is more appropriate for analyzing dedicated subnetworks or modules than large protein interaction networks, which are expected to contain such long cycles. We distinguish between two different types of problematic networks for our method. First type includes networks for which imposing a temporal order that encompasses all functional groups in the network is meaningless. Second type includes networks for which such order is meaningful, but the assumption that the overlap between functional groups has a tree-like structure is not valid. We plan to extend our approach to deal with networks of the second type by utilizing graph-theoretical tools developed for other specialized graph families, such as arc-intersection graphs.

Another issue that requires further investigation is the presence of noise in high-throughput protein interaction networks and its effect on the Tree of Complexes representation. While our method deals to some extent with false negatives, through its edge addition procedure, the issue of false positives is not addressed. We plan to explore alternative graph modification procedures that will incorporate both false negative and false positive interactions.

Acknowledgments

This work was supported by the intramural research program of the National Institutes of Health. Visualization of protein interaction networks was performed using the Pajek program [3]. The authors would like to thank anonymous reviewers for their constructive comments.

References

1. G.D. Bader and C.W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.
2. A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
3. V. Batagelj and A. Mrvar. Pajek - Program for large network analysis. *Connections*, 2:47–57, 1998.

4. T. Bouwmeester, A. Bauch, H. Ruffner, P.O. Angrand, G. Bergamini, K. Coughton, C. Cruciat, D. Eberhard, J. Gagneur, and S. Ghidelli. A physical and functional map of the human TNF-alpha/NF-kappaB signal transduction pathway. *Nature Cell Biology*, 6:97–105, 2004.
5. D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003.
6. D.G. Corneil, Y. Perl, and L. Stewart. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.
7. R.G. Downey and M.R. Fellows. *Parametrized Complexity*. Springer-Verlag, 1997.
8. E.A. Elion, M. Qi, and W. Chen. SIGNAL TRANSDUCTION: Signaling Specificity in Yeast. *Science*, 307(5710):687–688, 2005.
9. J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5(8):R57, 2004.
10. A.C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A.M. Michon, and C.M. Cruciat. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.
11. F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory (B)*, 16:47–56, 1974.
12. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
13. Y. Ho, A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, and K. Boutilier. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415:180–183, 2002.
14. K. Irie, M. Takase, K.S. Lee, D.E. Levin, H. Araki, K. Matsumoto, and Y. Oshima. MKK1 and MKK2, which encode *Saccharomyces cerevisiae* mitogen-activated protein kinase-kinase homologs, function in the pathway mediated by protein kinase C. *Molecular Cell Biology*, 13:3076–3083, 1993.
15. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences USA*, 98:4569–4574, 2001.
16. H.W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, 30:31–34, 2002.
17. N. Przulj, D.G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.
18. T.M. Przytycka and Y.K. Yu. Scale-free networks versus evolutionary drift. *Computational Biology and Chemistry*, 28:257–264, 2004.
19. A.W. Rives and T. Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences USA*, 100:1128–1133, 2003.
20. R. Shamir. Advanced topics in graph theory. Technical report, Tel-Aviv University, 1994.
21. V. Spirin and L.A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences USA*, 100:12123–12128, 2003.
22. S. Tornow and H. W. Mewes. Functional modules by relating protein interaction networks and gene expression. *Nucleic Acids Research*, 31:6283–6289, 2003.

23. P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, and P. Pochart. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403:623–627, 2000.
24. Y. Wang and H.G. Dohlman. Pheromone signaling mechanisms in yeast: a prototypical sex machine. *Science*, 306(5701):1508–1509, 2004.

Comparison of Protein-Protein Interaction Confidence Assignment Schemes

Silpa Suthram¹, Tomer Shlomi², Eytan Ruppin², Roded Sharan², and Trey Ideker¹

¹ Department of Bioengineering, University of California, San Diego, CA 92093, USA

² School of Computer Science, Tel-Aviv University, Israel
ssuthram@ucsd.edu

Abstract. Recent technological advances have enabled high-throughput measurements of protein-protein interactions in the cell, producing protein interaction networks for various species at an ever increasing pace. However, common technologies like yeast two-hybrid can experience high rates of false positive detection. To combat these errors, many methods have been developed which associate confidence scores with each interaction. Here we perform the first comparative analysis and performance assessment among these different methods using the fact that interacting proteins have similar biological attributes such as function, expression, and evolutionary conservation. We also introduce a new measure, the signal to noise ratio of protein complexes embedded in each network, to assess the quality of the different methods. We observe that utilizing any probability scheme is always more beneficial than assuming all observed interactions to be real. Also, schemes that assign probabilities to individual interactions generally perform better than those assessing the reliability of a set of interactions obtained from an experiment or a database.

1 Introduction

Systematic elucidation of protein-protein interaction networks will be essential for understanding how different behaviors and protein functions are integrated within the cell. Recently, the advent of high-throughput experimental techniques like yeast two-hybrid [1] assays and mass spectrometry [2] has led to the discovery of large-scale protein interaction networks in different species, including *S. cerevisiae* [2-5], *D. melanogaster* [6], *C. elegans* [7] and *H. sapiens* [8, 9]. Unfortunately, these large-scale data sets have so far been generally incomplete and associated with a significant number of false-positive interactions [10]. However, recent years have also seen an increase in the accumulation of other sources of biological data such as whole genome sequence, mRNA expression, protein expression and functional annotation. This is particularly advantageous as some of these data sets can be utilized to reinforce true (physical) protein interactions while downgrading others. For instance, true protein interactions have been shown to have high mRNA expression correlation for the proteins involved [11].

As a result, many bioinformatics approaches have been developed to unearth true protein interactions which can be mainly divided into two categories: (1) methods that assign reliability measurements to previously observed interactions; and (2) methods

that predict interactions *ab initio*. For category (1), Deane *et al.* [12] introduced one of the first methods to tackle the problem of assigning reliabilities to interactions using similarity in mRNA expression profiles. Subsequently, Bader *et al.* [13] and Deng *et al.* [14] used additional features of interacting proteins, including functional similarity and high network clustering [15], to assign confidence scores to protein interactions. For category (2), Marcotte *et al.* [16], von Mering *et al.* [17] and Jansen *et al.* [18] were among the first to predict new protein interactions by incorporating a combination of different features like high mRNA expression correlation, functional similarity, co-essentiality, and co-evolution. These schemes calculate a log-likelihood score for each interaction.

Here, we perform the first benchmarking analysis to compare the different interaction probability assignment schemes versus one another. We limit ourselves to methods that assign probabilities to interactions as opposed to those that compute a log-likelihood ratio. We also assess each of the methods against a “null hypothesis”, a uniform scheme which considers the same probability for all interactions. To compare the quantitative accuracy of the methods, we examine the correlations between the probability estimations and different biological attributes such as function and expression. As a further comparison criterion, we introduce and apply the signal processing concept of ‘Signal to Noise ratio’ (SNR) to evaluate the significance of protein complexes identified in the network based on the different schemes. The discovery of these complexes depends on the connectivity of the interaction network which is determined by the underlying interaction probability scheme [19]. Finally, we compare the different weighting schemes based on previous observations regarding the preference of interacting proteins to have similar conservation characteristics [20, 21].

2 Interaction Confidence Assignment Schemes

Although large-scale protein interaction networks are being generated for a number of species, *S. cerevisiae* (yeast) is perhaps the best studied among them and is associated with the largest variety and number of large-scale data. Hence, most of the interaction probability schemes have been developed specifically for the yeast protein interaction network. The yeast network was also the focus of our analysis in which we considered interaction probability scores by Bader *et al.* [13], Deane *et al.* [12], Deng *et al.* [14], Sharan *et al.* [19] and Qi *et al.* [22]. Bader *et al.*, Sharan *et al.* and Qi *et al.* assigned specific probabilities to every interaction, while Deane *et al.* and Deng *et al.* grouped the interactions into high/medium/low confidence groups. All of the above schemes estimated the predicted reliabilities of each interaction based on a gold standard set of positive and negative interaction data. Specifically, each weighting scheme defined gold standard sets based on various biological observations.

2.1 Bader et al. (BL / BH)

As a gold standard positive training data set, Bader *et al.* [13] used interactions determined by co-immunoprecipitation (co-IP), in which the proteins were also one or two links apart in the yeast two-hybrid (Y2H) network. The negative training data set

was selected from interactions reported either by co-IP or Y2H, but whose distance (after excluding the interaction) was larger than the median distance in Y2H or co-IP respectively. Using these training data, they constructed a logistic regression model which computes the probability of each interaction based on explanatory variables including data source, number of interacting partners, and other topological features like network clustering. We refer to this scheme as Bader *et al.* (low) or BL in our analysis.

Initially, the authors used measures based on Gene Ontology (GO) [24] annotations, co-expression, and presence of genetic interactions as measures to validate their data. However, they also combined these measurements into the probability score to bolster their confidence of true interactions. We consider these new confidence scores in our analysis as Bader *et al.* (high) or BH.

2.2 Deane *et al.* (DE)

Deane *et al.* [12] estimated the reliability of protein-protein interactions using the expression profiles of the interacting partners. Protein interactions observed in small-scale experiments and also curated in the Database of Interacting Proteins (DIP) [25] were considered as the gold standard positive interactions. As a gold standard negative, they randomly picked protein pairs from the yeast proteome that were not reported in DIP. The authors used this information to compute the reliabilities of groups of interactions (obtained from an experiment or a database). Higher reliabilities were assigned to groups whose combined expression profile was closer to the gold standard positive than the gold standard negative interactions. Specifically, reliabilities were assigned to the whole DIP database, the set of all protein interactions generated in any high-throughput genome screen, and protein interactions generated by Ito *et al.* [4]

2.3 Deng *et al.* (DG)

Deng *et al.* [14] estimated the reliabilities of different interaction data sources in a manner similar to Deane *et al.* [12]. They separately considered experiments that report pair-wise interactions like Y2H and those that report complex membership like mass spectrometry. Curated pair-wise interactions from the literature and membership in protein complexes from MIPS [23] were used as the gold standard positive set in each case. Randomly chosen protein pairs formed the gold standard negative data set. Reliabilities for each data source were computed using a maximum likelihood scheme based on the expression profiles of each data set. The authors evaluated reliabilities for Y2H data sources like Uetz *et al.* [5] and Ito *et al.* [4], and protein complex data sources like Tandem Affinity Purification (TAP) [2] and high-throughput mass spectrometry (HMS) [3]. In addition to assigning reliabilities to each dataset, the authors also provided a conditional probability scheme to compute probabilities for each interaction. We use the probabilities generated by this method for our comparative analysis.

2.4 Sharan *et al.* (SH)

Recently, we have also implemented an interaction probability assignment scheme [19] similar to the one proposed by Bader *et al.* The scheme assigned probabilities to interactions using a logistic regression model based on mRNA expression, interaction

3 Assessment of Interaction Schemes

As the probability schemes were previously computed for different subsets of yeast PPIs, we first compiled a set of 11,883 interactions common to all schemes. Five measures that have been shown to be associated with true protein interactions were used to assess the accuracy of each interaction probability scheme. In some cases, one of the measures used to assess a schemes' performance was already used as an input in assigning the probabilities. Although this creates some amount of circularity, the measure remains useful for gauging the performance of the remaining probability schemes. For each of the six measures, we evaluated two ways to estimate the level of association: Spearman correlation, and mutual information. The Equal probability scheme results in a spearman correlation and mutual information values of 0, by default. Consequently, we also evaluate the weighted average for each probability

scheme. The weighted average is given by $WA = \frac{\sum_{i=1}^N p_i * m_i}{\sum_{i=1}^N p_i}$, where p_i is the

probability of a given interaction and m_i is the value of one of the five measures for the interaction.

3.1 Global Properties of Interaction Probability Schemes

As a first measure, we compared global statistics such as the average and median probability assigned by each scheme and the number of interactions $p > 0.5$ (Table 2). We also computed Spearman correlations among the different probability schemes to measure their level of inter-dependency (Table 3). The maximum correlation was seen between BL and BH, as might be expected as both schemes were reported in the same study and BH was derived from BL. In addition, we also evaluated the spearman correlation between the measures used to assess accuracy of the probability schemes (see Appendix).

Table 2. Comparison of Global properties of different probability assignment schemes

Prob. Scheme	Average Probability	Median Probability	# Intr with prob > 0.5
BL	0.51	0.547	6,886
BH	0.477	0.496	5,896
DE	0.717	1	7,531
DG	0.39	0.25	4,799
SH	0.38	0.421	1,121
QI	0.97	0.99	11,658
AVG	0.574	0.671	7,866
EQ	0.99	0.99	11,883

Table 3. Correlation of different probability schemes. The p-values for all the correlation measurements were very significant (p -value $2e-16$).

	BH	DE	DG	SH	QI
BL	0.923	0.655	0.633	0.626	0.371
	BH	0.672	0.644	0.665	0.416
		DE	0.718	0.847	0.238
			DG	0.68	0.466
				SH	0.274

3.2 Gene Ontology (GO) Similarity

As a second measure, we adopted the common notion that two interacting proteins are frequently involved in the same process and hence should have similar GO assignments [24]. The Gene Ontology terms are represented using a directed acyclic graph data structure in which an edge from term ‘a’ to term ‘b’ indicates that term ‘b’ is either a more specific functional type than term ‘a’, or is a part of term ‘a’. As a result, terms that come deeper in the graph are more specific. Moreover, specific terms also have less number of proteins assigned to them. Hence, we evaluated the size (number of proteins assigned to the term) of the deepest common GO term assignment (deepest common ancestor) shared between a pair of proteins that interact. The gene ontology annotations for yeast proteins were obtained from the July 5th, 2005 download from Saccharomyces Genome Database (SGD) [26] and the association between terms were obtained from the Gene Ontology consortium (<http://www.geneontology.org/>).

Table 4. Association of interaction probabilities with the size of the deepest common ancestor in the Gene Ontology assignments and mRNA expression correlation. Shaded cells indicate schemes which used similar GO annotation or mRNA expression profiles as an input to assigning interaction reliability. p -values for all the correlation measurements were very significant (p -value $2e-16$). SC: Spearman Correlation; MI: Mutual Information; WA: Weighted Average.

Prob. Scheme	GO Annotation			Expression Correlation		
	SC	MI	WA	SC	MI	WA
BL	-0.42	0.16	5.85	0.185	0.0531	0.494
BH	-0.5	0.22	5.68	0.223	0.0626	0.503
DE	-0.385	0.07	5.91	0.016	0	0.481
DG	-0.49	0.17	5.62	0.185	0.041	0.511
SH	-0.47	0.157	5.71	0.05	0.012	0.492
QI	-0.444	0.013	6.34	0.337	0	0.481
AVG	-0.545	0.26	5.93	0.205	0.08	0.585
EQ	—	—	6.32	—	—	0.482

Table 4 shows the relationship between the size of the deepest common GO term and interaction probabilities for each scheme. The probabilities generated by BH have relatively high correlation with the GO term assignments. This result is not surprising since gene ontology assignments are taken as input to probability calculation in this scheme. Although QI has a relatively high Spearman correlation coefficient, its weighted average and mutual information values are the worst.

3.3 Presence of Conserved Interaction in Other Species (Interologs)

Presence of conserved interactions across species is believed to be associated with biologically meaningful interactions [27]. However, since most species' interaction networks are still incomplete, it is important not to skew the results of this analysis due to false-negatives. As our benchmark, we used yeast protein interactions that were conserved with measured *C. elegans* (worm) and *D. melanogaster* (fly) interactions obtained from the Database of Interacting Proteins (DIP). An interaction was considered conserved if the orthologs of the interacting proteins were also interacting. Putative orthologs were assigned based on sequence similarity computed using BLAST [28]. We evaluated the weighted average between the probability assignment for each yeast interaction and the number of conserved interactions across worm and fly (0, 1, or 2) and repeated the analysis for different BLAST E-value thresholds for homology assignments (Fig. 1). At higher E-value thresholds we observe that SH has the highest weighted average but has similar values to BL and BH at lower thresholds. QI has similar weighted averages to Equal, and both consistently have lower values than the remaining schemes.

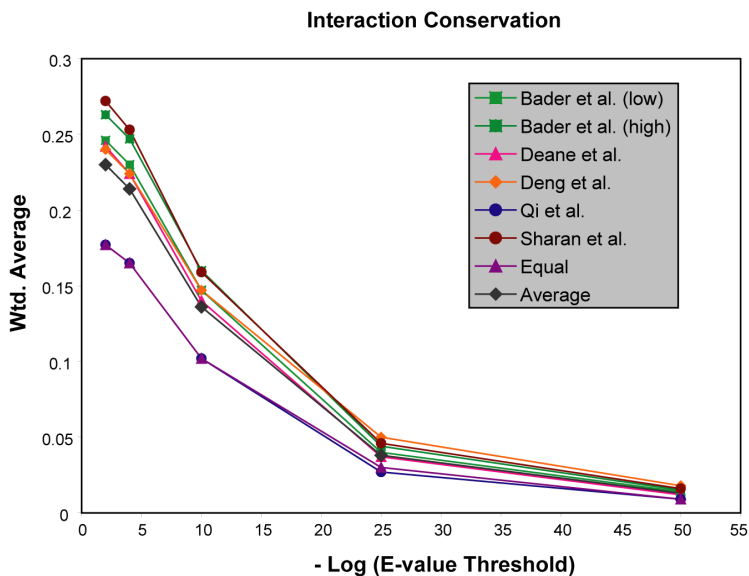


Fig. 1. Correlation of number of conserved interactions and probability assignments to interactions

3.4 Expression Correlation

Yeast expression data for ~790 conditions were obtained from the Stanford Microarray Database (SMD) [29]. For every pair of interacting proteins, we computed the Pearson correlation coefficient of expression. We then calculated the Spearman correlation, mutual information and weighted average between the expression correlation coefficient of interacting proteins and their corresponding probability assignments in the different schemes (see Table 4). We found significant association between expression correlations and probabilities in the case of BH, BL, QI and DG. This result is expected as these schemes, with the exception of BL, utilize expression similarity for interaction probability calculation. Surprisingly, DE probabilities showed very little correlation with expression, even though mRNA expression profiles were used as input in the prediction process. On the other hand, BL had a higher Spearman correlation and mutual information than SH, though they had very similar weighted averages and SH did not utilize expression data in the training phase.

3.5 Signals to Noise Ratio of Protein Complexes

Most cellular processes involve proteins that act together in pathways or complexes. Recently, several methods [19, 30-33] have been developed to ascertain the biologically meaningful complexes encoded within protein interaction networks. These methods search for complexes modeled as dense protein interaction sub-networks. Here, we applied a previously published algorithm [19] to discover complexes in the yeast network. We evaluated the resulting complexes using signal to noise ratio [34]. Signal to noise ratio (SNR) is a standard measure used in information theory and signal processing to assess data quality.

To compute SNR, a search for dense interaction complexes is initiated from each node (protein) and the highest scoring complex from each is reported. This yields a distribution of complex scores over all nodes in the network. A score distribution is also generated for 100 randomized networks which have identical degree distribution to the original network [35]. The randomized versions of interaction networks were generated by randomly reassigning the interactions, while maintaining the number of interactions per protein. SNR ratio is computed using these original and random score distributions (representing signal and noise, respectively) according to the standard formula [36] using the root-mean-square (rms):

$$\text{SNR} = \log_{10} \frac{\text{rms}(\text{original complex scores})}{\text{rms}(\text{random complex scores})}, \quad \text{where } \text{rms}(x) = \sqrt{\frac{1}{M} \sum_{i=1}^M x_i^2}$$

As the SNR is independent of the number of complexes reported, we can directly compare its value across the different probability schemes (Table 5). Here, DE and EQ probabilities have low SNR, while SH and DG have the highest SNR values.

Table 5. Associations of conservation rate coherency scores and SNR with interaction probabilities. SC: Spearman Correlation. Note that conservation scores based on weighted averages and mutual information were omitted as they were similar across the different weighting schemes.

Prob. Scheme	Conservation Coherency (SC)	SNR
BL	-0.09	0.734
BH	-0.104	0.735
DE	-0.113	0.537
DG	-0.141	0.95
SH	-0.126	0.742
QI	-0.12	0.72
AVG	-0.132	0.73
EQ	—	0.657

Table 6. Ranking of the probability schemes in the five measures used for assessing their quality. Schemes with rank 1 perform the best. SC: Spearman Correlation; WA: Weighted Average; SNR: Signal to Noise Ratio. The shaded boxes indicate the measures used as input for the corresponding probability scheme.

Probability Scheme	Gene Ontology (SC/WA)	Interaction Conservation (WA)	Gene Expression (SC/WA)	SNR	Conservation Coherency (SC)	Average Rank
Bader <i>et al.</i> (low)	6 / 4	3	4 / 4	4	7	4.14
Bader <i>et al.</i> (high)	2 / 2	2	2 / 3	3	6	3.66
Deane <i>et al.</i>	7 / 5	4	6 / 6	8	5	5.8
Deng <i>et al.</i>	3 / 1	4	4 / 2	1	1	2
Sharan <i>et al.</i>	4 / 3	1	5 / 5	2	3	3.28
Qi <i>et al.</i>	5 / 8	6	1 / 6	6	4	5.33
Average	1 / 6	5	3 / 1	5	2	3.28
Equal	- / 7	6	- / 6	7	-	6.5

3.6 Evolutionary Conservation

Interacting proteins show a clear preference to be conserved as a pair, indicating a selection pressure on the interaction links between proteins [20]. For every pair of interacting proteins, we computed the conservation rate coherency score as the absolute value of the difference between the evolutionary rates of the two corresponding genes. Low scores indicate highly coherent conservation rates. Evolutionary rates were obtained from Fraser *et al.* [21], estimated using nucleotide substitution rates. We then calculated the Spearman correlation between the conservation rate coherency scores of

interacting proteins and their corresponding probability assignments in the different schemes (see Table 6). For all probability assignment schemes we obtained a statistically significant negative correlation (p-value < 0.05) between the conservation rate discrepancy scores and the corresponding probabilities, indicating that proteins with high probability interactions tend to have similar conservation rates. The highest correlation (in absolute value) was obtained for DG.

4 Discussion

In summary, we have compared six of the available schemes that assign confidence scores to yeast interactions with each other and also with a uniform scheme. Table 6 gives the relative ranking of these schemes over the five measures used to assess their reliability.

Firstly, we find that EQ almost always ranks the lowest, suggesting that utilizing a probability scheme is always more beneficial than considering all observed interactions to be true. Secondly, QI has comparable ranks to other schemes when considering Spearman correlation coefficient, but generally has very low ranks when considering weighted average.

We conjecture that this trend is influenced by the relatively small standard deviation in the estimated probabilities in that scheme which assigns high probabilities to all interactions with (with 11447 interactions above 0.9) Thirdly, Deane *et al.* is the only scheme which assigns reliabilities to a set of interactions rather than individual interactions and generally performs poorly compared to other interactions schemes (Table 6). This suggests that probability schemes assessing the quality of each interaction by itself are more reliable.

We calculated the average ranks for each probability assignment schemes. To avoid circularity, the average ranks were computed by considering only those measures which were not used as input for the scheme in question. Overall, Deng *et al.* performs the best and Sharan *et al.* and the average scheme follow it as a close second.

Acknowledgments

We gratefully acknowledge the following funding support for this research: a National Science Foundation Quantitative Systems Biology grant (SS, TI); the National Institute of General Medical Sciences (GM070743-01, TI); a David and Lucille Packard Fellowship award (TI); the Alon fellowship (RS); and the Tauber Fund (TS).

References

1. Fields, S. and O. Song, *A novel genetic system to detect protein-protein interactions.* Nature, 1989. **340**(6230): p. 245-6.
2. Gavin, A.C., et al., *Functional organization of the yeast proteome by systematic analysis of protein complexes.* Nature, 2002. **415**(6868): p. 141-7.

3. Ho, Y., et al., *Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry*. Nature, 2002. **415**(6868): p. 180-3.
4. Ito, T., et al., *A comprehensive two-hybrid analysis to explore the yeast protein interactome*. Proc Natl Acad Sci U S A, 2001. **98**(8): p. 4569-74.
5. Uetz, P., et al., *A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae*. Nature, 2000. **403**(6770): p. 623-7.
6. Giot, L., et al., *A protein interaction map of Drosophila melanogaster*. Science, 2003. **302**(5651): p. 1727-36.
7. Li, S., et al., *A map of the interactome network of the metazoan C. elegans*. Science, 2004. **303**(5657): p. 540-3.
8. Stelzl, U., et al., *A human protein-protein interaction network: a resource for annotating the proteome*. Cell, 2005. **122**(6): p. 957-68.
9. Rual, J.F., et al., *Towards a proteome-scale map of the human protein-protein interaction network*. Nature, 2005. **437**(7062): p. 1173-8.
10. von Mering, C., et al., *Comparative assessment of large-scale data sets of protein-protein interactions*. Nature, 2002. **417**(6887): p. 399-403.
11. Grigoriev, A., *A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage T7 and the yeast Saccharomyces cerevisiae*. Nucleic Acids Res, 2001. **29**(17): p. 3513-9.
12. Deane, C.M., et al., *Protein interactions: two methods for assessment of the reliability of high throughput observations*. Mol Cell Proteomics, 2002. **1**(5): p. 349-56.
13. Bader, J.S., et al., *Gaining confidence in high-throughput protein interaction networks*. Nat Biotechnol, 2004. **22**(1): p. 78-85.
14. Deng, M., F. Sun, and T. Chen, *Assessment of the reliability of protein-protein interactions and protein function prediction*. Pac Symp Biocomput, 2003: p. 140-51.
15. Goldberg, D.S. and F.P. Roth, *Assessing experimentally derived interactions in a small world*. Proc Natl Acad Sci U S A, 2003. **100**(8): p. 4372-6.
16. Marcotte, E.M., et al., *A combined algorithm for genome-wide prediction of protein function*. Nature, 1999. **402**(6757): p. 83-6.
17. von Mering, C., et al., *STRING: known and predicted protein-protein associations, integrated and transferred across organisms*. Nucleic Acids Res, 2005. **33**(Database issue): p. D433-7.
18. Jansen, R., et al., *A Bayesian networks approach for predicting protein-protein interactions from genomic data*. Science, 2003. **302**(5644): p. 449-53.
19. Sharan, R., et al., *Conserved patterns of protein interaction in multiple species*. Proc Natl Acad Sci U S A, 2005. **102**(6): p. 1974-9.
20. Pagel, P., H.W. Mewes, and D. Frishman, *Conservation of protein-protein interactions - lessons from ascomycota*. Trends Genet, 2004. **20**(2): p. 72-6.
21. Fraser, H.B., et al., *Evolutionary rate in the protein interaction network*. Science, 2002. **296**(5568): p. 750-2.
22. Qi, Y., J. Klein-Seetharaman, and Z. Bar-Joseph, *Random forest similarity for protein-protein interaction prediction from multiple sources*. Pac Symp Biocomput, 2005: p. 531-42.
23. Mewes, H.W., et al., *MIPS: a database for protein sequences, homology data and yeast genome information*. Nucleic Acids Res, 1997. **25**(1): p. 28-30.
24. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium*. Nat Genet, 2000. **25**(1): p. 25-9.
25. Xenarios, I., et al., *DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions*. Nucleic Acids Res, 2002. **30**(1): p. 303-5.

26. Christie, K.R., et al., *Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms*. Nucleic Acids Res, 2004. **32**(Database issue): p. D311-4.
27. Yu, H., et al., *Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs*. Genome Res, 2004. **14**(6): p. 1107-18.
28. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucleic Acids Res, 1997. **25**(17): p. 3389-402.
29. Ball, C.A., et al., *The Stanford Microarray Database accommodates additional microarray platforms and data formats*. Nucleic Acids Res, 2005. **33**(Database issue): p. D580-2.
30. Bader, G.D. and C.W. Hogue, *An automated method for finding molecular complexes in large protein interaction networks*. BMC Bioinformatics, 2003. **4**: p. 2.
31. Hu, H., et al., *Mining coherent dense subgraphs across massive biological networks for functional discovery*. Bioinformatics, 2005. **21 Suppl 1**: p. i213-i221.
32. Kelley, B.P., et al., *Conserved pathways within bacteria and yeast as revealed by global protein network alignment*. Proc Natl Acad Sci U S A, 2003. **100**(20): p. 11394-9.
33. Spirin, V. and L.A. Mirny, *Protein complexes and functional modules in molecular networks*. Proc Natl Acad Sci U S A, 2003. **100**(21): p. 12123-8.
34. Suthram, S., Sittler, T., Ideker, T., *The Plasmodium protein network diverges from those of other eukaryotes*. NatureDOI : 10.1038/nature04135, 2005.
35. Canfield, B., 1978.
36. Shanmugan, K.S., *Digital and analog communication systems*. 1979, New York: Wiley. xviii, 600.

Appendix

Table A1. Spearman correlation of different measurement schemes. The p-values for the correlation measurements were very significant (p-value $2e-16$).

	Interaction Conservation	Conservation Coherency	Expression Correlation
Gene Ontology	-0.14	0.14	-0.287
	Interaction Conservation	-0.08	0.058
		Conservation Coherency	-0.11

Characterization of the Effects of TF Binding Site Variations on Gene Expression Towards Predicting the Functional Outcomes of Regulatory SNPs

Michal Lapidot and Yitzhak Pilpel

Department of Molecular Genetics, Weizmann Institute of Science,
Rehovot, 76100, Israel
michal.lapidot@weizmann.ac.il, pilpel@weizmann.ac.il
<http://longitude.weizmann.ac.il/>

Abstract. This work addresses a central question in medical genetics – the distinction between disease-causing SNPs and neutral variations. Unlike previous studies that focused mainly on coding SNPs, our efforts were centered around variations in regulatory regions and specifically within transcription factor (TF) binding sites. We have compiled a comprehensive collection of genome wide TF binding sites and developed computational measures to estimate the effects of binding site variations on the expression profiles of the regulated genes. Applying these measures to binding sites of known TFs, we were able to make predictions that were in line with published experimental evidence and with structural data on DNA-protein interactions. We attempted to generalize the properties of expression-altering substitutions by accumulating statistics from many substitutions across multiple binding sites. We found that in the yeast genome substitutions that abolish a G or a C are on average more severe than substitutions that abolish an A or a T. This may be attributed to the low GC content of the yeast genome, in which G and C may be important for conferring specificity. We found additional factors that are correlated with the severity of a substitution. Such factors can be integrated in order to create a set of rules for the prioritization of regulatory SNPs according to their disease-causing potential.

1 Introduction

The identification of disease-causing mutations is a central objective of medical genetics. So far most efforts to distinguish disease-causing single nucleotide polymorphisms (SNPs) from neutral variations have focused on coding SNPs [1-7]. Regulatory region variants are also known to cause diseases through altering the expression profiles of their downstream genes [8, 9]. Estimates show that the human population contains thousands of *cis*-regulatory variations [10]. Such high numbers set a clear need for the development of computational means for the identification of potentially deleterious regulatory SNPs. The present work lays the foundations for the development of such tools. It does not yet address actual SNPs, but rather studies a comprehensive collection of genome wide transcription factor binding sites (TFBS), in order to characterize the effects of binding site variations on the expression profiles of the regulated genes. TFBS

are short (~6-20 bases) redundant sequences. Transcription factors (TFs) recognize a range of binding sites that may differ at several positions. Substituting a single position within a binding site may thus, in many cases maintain the site in the recognition domain of the same TF. There is however a possibility that the substitution will result in binding site loss or in the acquisition of a binding site recognized by another TF (Figure 1 left panel). The aim of this work was to develop computational methods for distinguishing between these three possible scenarios without the need to systematically mutate each binding site. These methods were first applied to individual binding sites, and the results were generalized to deduce universal properties of expression-altering substitutions. We focused here on the *Saccharomyces cerevisiae* (*SC*) genome because vast knowledge already exists regarding TFBS in this organism; however the presented work can be easily applied to other organisms and specifically to human.

As a first step, we have compiled a comprehensive dataset of TFBS in the *SC* genome ([11], Lapidot et al in prep.). Each binding site was defined by its DNA sequence (its syntax) and assigned a likely regulatory function (its semantics), in terms of the expression profiles of the genes it controls and the experimental conditions in which it operates. Our set had a good coverage of a recently published TFBS set ([12]), as well as many novel binding sites (see [11] for more details). An analysis of this binding site collection revealed a non trivial relation between syntax and semantics: Binding sites with similar syntax may yield different expression patterns of the regulated genes, or operate at different conditions (differ on the semantic level), whereas binding sites with different syntax can have similar semantics (i.e. dictate similar expression patterns). We next developed computational measures to estimate the semantic consequence of substituting a single binding site position. We applied these measures to binding sites of known TFs and were able to make predictions that were in line with published experimental evidence and with structural data on DNA-protein interactions. We further attempted to generalize the properties of expression-altering substitutions by accumulating statistics from many substitutions across multiple binding sites. Finally we tested out additional factors that are correlated with the severity of a substitution, such as the Information Content (IC) of the substituted position. These factors can be integrated to form a prioritization scheme that will enable the prediction of potentially deleterious regulatory SNPs.

2 Results

2.1 Compiling a Comprehensive TFBS Collection

This study was conducted in the *SC* genome, for which vast TFBS knowledge already exists. However in order to both broaden this knowledge and form a quantifiable connection between binding site sequence and the expression profiles of the regulated genes, we compiled our own comprehensive dataset. This dataset is unbiased by prior knowledge and is based on the premise that any nucleotide sequence that resides in a promoter of a gene may contribute to its expression regulation. We applied the previously described Expression Coherence (EC) score [13-15] to assess the effect of a promoter sequence motif on the related gene's expression profile. The EC score measures the extent to which a set of genes (in this case the set is defined by a common motif sequence) display similar expression profiles at a given condition.

The dataset was produced by integrating whole genome SC promoter sequences with expression patterns of the corresponding genes in 40 experimental conditions including cell cycle, sporulation and various stress responses (see http://longitude.weizmann.ac.il/TFLocation/conditions_explist.html for a full list of conditions). We systematically scanned all k-mers (k ranges from 7-11) that appear in SC promoters. For each k-mer, we computed the EC score of the set of genes that contain it in their promoter across the 40 conditions. A p-value was assigned to each EC score, which estimates the probability of obtaining the observed or higher EC score by chance [15] and false discovery rate (FDR) [16] of 0.1 was applied to correct for multiple hypotheses. A total of 8610 sequence motifs appeared significant in at least one of the tested experimental condition. These comprise the core of the dataset (hereafter referred to as the ‘core dataset’).

2.2 Method Validation and Comparison to Published Datasets

To validate the ability of our method to identify biologically significant motifs we tested out whether previously published regulatory motifs score highly using the same method. 89/102 (87%) of the TF binding sites published by Harbison et al. [12] passed FDR of 0.1 in at least one experimental condition, and thus would have been discovered by our method. For comparison only 15/102 (15%) random gene sets (identical in size to the gene sets containing each of Harbison’s motifs) appeared significant in at least one condition. Additionally we assessed our coverage of Harbison’s set by comparing each core motif to all of Harbison’s positional weight matrices (PWMs) [11]. We devised a score between 0-100% that denotes how likely a given string is to be generated from a given PWM (see methods). Requiring a match score of 99% we obtain a coverage of 89/102 (87%), and of 70/77 (91%) of the non redundant Harbison set (The 102 PWMs fall into clusters of highly similar motifs). By relaxing the similarity requirement to 90% the coverage increases to 100/102 motifs, falling into 76/77 distinct clusters (table 1).

Table 1. Coverage of the Harbison motif set by our dataset. A scoring method was devised to assess how likely a given string is to be generated from a given PWM. The score is on a scale of 0 to 100 (see methods). We computed this score for all 8610 core motifs over the 102 Harbison PWMs. The coverage of Harbison’s motif set was assessed for several different score cutoffs. Note that a single string may match more than one Harbison PWM, because of redundancy in Harbison’s dataset. There are two very long (17 and 18 positions) gapped motif in Harbison’s set, for which we have no match, because our set only covers motifs of length 7-11.

Score Cut-off	Coverage of our dataset	Coverage of Harbison	Coverage of unique Harbison clusters
99%	1402/8610=16%	89/102=87%	70/77=91%
98%	1528/8610=18%	93/102=91%	73/77=95%
97%	1719/8610=20%	96/102=94%	73/77=95%
95%	2198/8610=25%	99/102=97%	75/77=97%
92%	3251/8610=38%	99/102=97%	75/77=97%
90%	4161/8610=48%	100/102=98%	76/77=99%

2.3 Exploiting Our Dataset to Predict the Outcome of a Binding Site Substitution

In the process of producing our core dataset, we assigned EC scores, corresponding p-values and likely expression effects to all k-mers residing in yeast promoters, regardless of whether they were ultimately included in the dataset. This provided a unique source of information for our analysis: By comparing the EC scores and the induced expression profiles of k-mers differing in a single position we could predict the outcome of a substitution that transforms one k-mer into the other. Three main scenarios were observed (i) Two k-mers differing in a single position both belong to the core dataset (passed FDR) and regulate genes with a similar expression profile. This implies that the k-mers are recognized by the same TF and a substitution from one to the other will have a very mild effect (Figure 1, green arrows). (ii) The two k-mers belong to the core dataset but regulate genes with a different expression profile. This may imply that they are recognized by different TFs, thus a substitution from one k-mer to the other will cause binding site switching (Figure 1, blue arrows). (iii) One k-mer belongs to the core set whereas the other did not pass the FDR constraint. This implies that substituting the former to the latter will result in binding site loss without acquisition of a new site (Figure 1 red arrows).

We devised three quantitative measures in order to compare the regulatory functions of two k-mers: (1) ΔEC – the difference in EC scores between the set of genes containing k-mer a in their promoters and the set of genes containing k-mer b in their promoters. (2) ΔPV – the difference in the logarithm of p-values assigned to the EC scores of the two gene sets (3) Distance in expression profiles – each k-mer is represented by the mean expression profile of all genes containing it in their promoters. This measure is the distance between the mean expression profiles of the two gene sets (calculated as 1-correlation coefficient of the two vectors representing the means).

Note that although these three measures may seem redundant, they capture slightly different phenomena; An unaltered EC score (low ΔEC) accompanied by a significant change in mean expression profile may imply TF switching, whereas the opposite case in which the expression profile is maintained, but there is a decrease in coherence (high ΔEC) may imply lower affinity to the same TF. The combination of these two measures could thus aid in differentiating between cases of TF switching and cases of a reduction in binding affinity of the same TF.

We have developed a computational tool termed ‘motif landscape analysis’ [15] that employs our comprehensive dataset in order to systematically predict the outcome of all possible single nucleotide substitutions within a given motif. For a motif of length L this tool examines all 3^*L k-mers that are obtained by substituting the motif consensus at each single position. For each such k-mer it computes the three described measures ΔEC , ΔPV and distance in expression profiles between genes containing it in their promoters and genes containing the consensus motif. The results are graphically displayed using a modified version of the previously introduced Combinogram [13] showing the EC scores of the gene set including each of the 3^*L motif variants in their promoters and the similarity of their averaged expression profiles.

Applying this tool to the yeast sporulation factor Ndt80 (Figure 1 right panel) using the *SC* sporulation expression data, predicted that two out of the three possible substitutions in the second position will have only a minor effect on expression whereas an A->G substitution at the same position will result in a severe effect (see figure 1

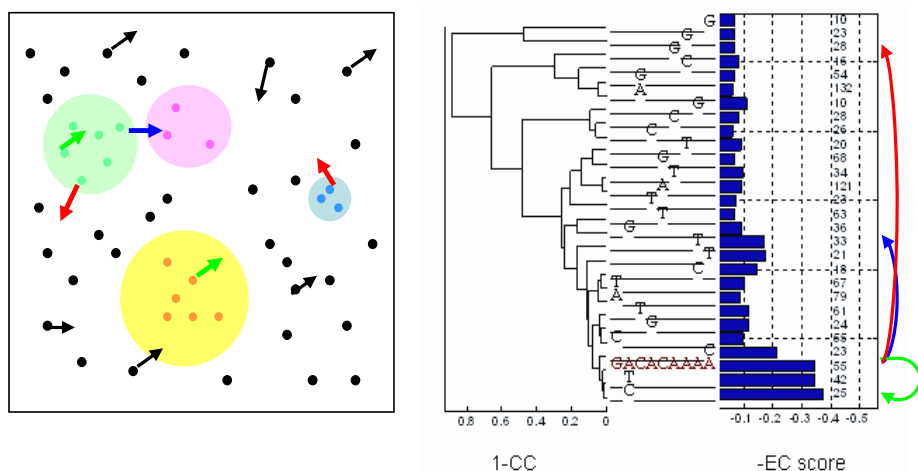


Fig. 1. Possible outcomes of binding site substitutions: Left panel a cartoon depicting possible effects of mutations in regulatory motifs. Points represent promoter elements and discs represent transcription factor recognition ranges. Points that are included within the disc of a given TF represent promoter elements that are bound by the TF. Arrows illustrate the result of single nucleotide substitution within a promoter element. Such a substitution, can cause binding site loss (*red arrows*), a change in affinity to the same TF (*green arrows*), or binding site switching - creation of a binding site with higher affinity to a different TF (*blue arrows*). The right panel illustrates the detection of the same outcomes using our motif landscape analysis tool (as described in detail in [15]). This display captures the effects of single nucleotide substitutions of a given motif on the expression profiles of the downstream genes. The analyzed motif is the yeast Ndt80 sporulation factor (*wild type motif marked in red*). The dendrogram on the left part of the display shows the similarity in mean expression profiles between gene sets bearing variations of the motif in their promoters. The right side of the display shows the similarity within sets of genes that contain the same motif variation in their promoters, as measured by the EC score (the numbers correspond to the gene set sizes). The middle section displays the sequence of the motif variation studied in the corresponding row (with a ‘-’ indicating same nucleotide as the wild type motif). A substitution, that is in the recognition range of the same TF, is expected to maintain a high EC score and a similar expression profile (*green arrow*), A substitution that causes binding site loss, is expected to be recognized by both loss of coherence and a change in the mean expression profile (*red arrow*). A substitution that creates a new motif, that is in the recognition range of a different TF, is expected to maintain high expression coherence, while altering the mean expression profile (*blue arrow*). The second motif position appears relatively tolerant to substitutions, 2 out of the 3 possible single nucleotide substitutions of this position do not alter TF recognition (green substitutions). This observation is supported by the recently published structural data of Ndt80 bound to DNA [17]. The second motif position does not form a contact with the protein¹.

legend for details). When averaging over all possible single nucleotide substitutions, the second position appears to be the most tolerant towards substitutions and the seventh position - the most sensitive (figure 2). These results are in agreement with the

¹ The figures of this paper appear in color in the online version of the book.

structural data of Ndt80 bound to its DNA target [17]; the second ‘permissive’ motif position is the only position which does not form a direct contact with the protein. It is also supported by recently published *in vivo* reporter expression experiments and *in vitro* binding assays of Ndt80 mutants, that showed that this position is the most permissive one, and that, as predicted here, G is the only nucleotide that when placed at this position weakens binding affinity and reduces expression level of the reporter gene [18]. This implies that our method can complement and in some cases replace time consuming mutation experiments.

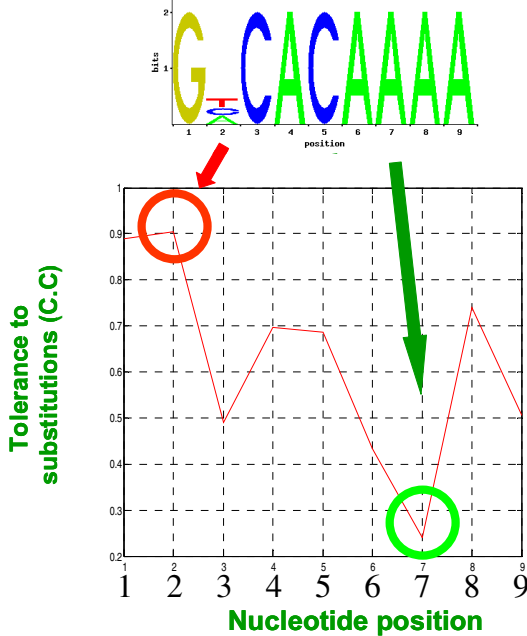


Fig. 2. The averaged tolerance to substitution for each nucleotide position within the Ndt80 motif was defined as the averaged correlation coefficient between the averaged expression profiles of the genes that have a perfect match to the consensus motif and the averaged expression profiles of the genes that have each of the three possible substitutions relative to the consensus in that position

2.4 Deducing General Properties of Expression-Altering Substitutions

Encouraged by our ability to predict the effects of binding site substitutions within a single motif, we attempted to generalize these predictions in order to define universal properties of substitutions that alter gene expression. We used the three measures described above to assess the severity of a substitution from base i to base j in a regulatory motif. Namely: change in EC score, change in p-value and change in mean expression profiles of genes regulated by a motif with nucleotide j versus genes regulated by the same motif with nucleotide i at the substituted position. This time, instead of analyzing a single motif we accumulated statistics from substitutions of different positions across multiple binding sites. There are twelve possible single nucleotide substitutions from base i to base j (when i can be A,C,G or T, and $j \neq i$). Each

severity measure was averaged over all substitutions of the type $n_i \rightarrow n_j$ in any possible motif. The motifs used for this analysis were core dataset motifs that correspond to known TFBS from Harbison's set [12].

Our first question was whether there were substitution types that are more radical than others (in analogy to amino acid substitutions where there are conservative changes that maintain the chemical properties of the residue versus radical changes that form a residue with different characteristics). Interestingly, although there was no single substitution type that appeared more radical than others, there were systematically higher penalties for substitutions that abolished a C or a G versus substitutions that abolished an A or a T (figure 3). Because the yeast genome is AT rich, this result may suggest that C and G are the nucleotides that confer specificity to a motif, and thus their substitution bears a greater effect on the motif's function. This raises a prediction that in other genomes with different GC content of the regulatory regions the penalties might be different, reflecting loss of information content with the elimination of different nucleotides. We intend to check this hypothesis in the human genome, which has a higher GC content than yeast, using the tools developed here.

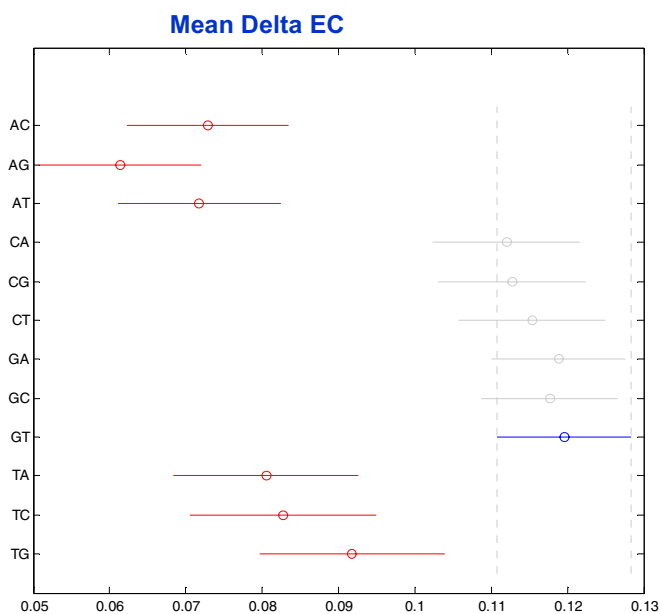


Fig. 3. Effects of each of the twelve possible single nucleotide substitutions on expression. The data was accumulated from all possible substitutions of each type in a dataset of highly scoring k-mers that correspond to known Harbison PWMs. The 'severity measure' applied is mean delta EC, thus high values correspond to severe substitutions. A clear trend is seen whereby substitutions that abolish an A or a T are less severe than substitutions that abolish a C or a G.

2.5 The Information Content of the Substituted Position

The degree of conservation of the substituted position in the PWM may also affect the severity of the phenotype. This was shown to be the case in protein coding SNPs [1, 2],

but parallel investigations were not carried out in promoter motifs. Substitutions of highly conserved positions are expected to have a more dramatic effect on expression compared to substitutions of positions with low conservation. To test this hypothesis we analyzed high scoring k-mers from our dataset which correspond to known Harbison PWMs. For these motifs both the expression measures obtained in the process of creating our dataset (EC, p-value, expression profiles) and the information content (IC) of all PWM positions are available. We could thus assess the correlation between the IC of a position and its sensitivity to substitution based on the previously described severity measures. Indeed a significant correlation exists between the mean expression distance and the IC of a position (table 2). The mean expression distance is also highly correlated to our other two expression based measures mean ΔEC and mean ΔPV .

Table 2. Correlations between the three expression measures mean ΔEC , mean ΔPV , mean expression distance and the IC of a PSSM position. Data was accumulated for 1867 positions. In each table cell, the first number is the correlation and the second number is the p-value on this correlation. The different expression measures are highly correlated. There is a correlation between the measure Mean Expression distance and the information content of a position. Note that the change in mean expression profile has a very significant but rather low correlation (0.3) with the other two expression based measures. This is because there are cases where high EC is maintained, but the expression profile changes (implying TF switching), and cases where the expression profile is maintained, but there is a decrease in coherence (implying lower affinity to the same TF).

	Mean ΔEC	Mean ΔPV	Mean expression Distance	Position IC
Mean ΔEC	1			
Mean ΔPV	0.5402 6.12e-142	1		
Mean Expression Distance	0.3505 4.34e-055	0.3053 1.41e-041	1	
Position IC	0.0827 3.47e-004	0.0531 0.0217	0.1252 5.7785e-008	1

3 Discussion

We have composed a comprehensive dataset of TFBS and developed measures for quantifying the effect of a binding site, present in the promoter, on the expression profiles of the regulated genes. These measures allowed us to compare the effects on gene expression of binding sites differing by a single nucleotide position, and to infer from the comparison what would be the severity of substituting one binding site into the other. We applied our tools to the yeast genome and were able to produce reliable predictions about the outcome of single nucleotide substitutions in a single binding site. By accumulating statistics for many substitutions across multiple binding sites we observed that not all nucleotide substitutions are similar in severity: In the *SC* genome abolishing a C or a G has a harsher effect on average than abolishing an A or a T. Although this result may be specific to the AT rich *SC* genome, the same measures

and tools can be easily applied to other genomes, and specifically to human. We have showed that other characteristics of a substituted motif position, such as its IC are in correlation with our measures of the effect on expression. We intend to test additional features including the evolutionary conservation of the substituted position and its vicinity to the protein in the DNA-protein co-crystal structure. All these features can be integrated to form a prioritization scheme that would allow the ranking of existing genome variations by their disease-causing potential.

The approach presented here demonstrates for the first time how a huge amount of data, on all known yeast TFs, using all genes, whose expression was monitored in multiple conditions, can be harnessed and utilized for taking the first step towards assessing the effects of nucleotide substitutions in TF binding sites. A conceptual analog of this endeavor for assessing the effects of amino acid substitutions on protein function could amount to mutating many proteins, say enzymes, in many different ways, and checking for each mutation reduction, or change, in biochemical activity and specificity. Since data for such effort is not even close to become available, the methodology presented utilizes in a unique way data that is available for its domain. While the main advantage of our methodology is the huge sample size, the disadvantage is that we are unable to control for other differences between promoters of analyzed genes (i.e. differences that are outside of the substituted position). The fact that we get statistically significant differences between the effects of different types of substitutions on expression likely indicates that despite uncontrolled sources of variation we extracted genuine signals.

An additional application of the present approach may be in algorithms that assign PWMs to promoters (e.g. PRIMA [19]) as it should provide means to weigh differently mismatches between the PWM preferences and the promoter sequence based on expected effect on expression.

4 Materials and Methods

4.1 Dataset Construction

Promoter sequences for 5651 *SC* genes were taken from SGD [20]. Expression data for 40 different time series experiments was downloaded from ExpressDB [21]. The promoters were systematically scanned for all occurrences of every possible k-mer (k varies from 7-11), resulting in an index file listing for each k-mer the set of genes that contain it in their promoters, along with the positions and orientations (strand). For the purpose of indexing each k-mer was combined with its reverse complement because it is well accepted that TFs bind double stranded DNA.

Following the k-mer indexing, EC scores in various experimental conditions were calculated for the sets of genes containing each of the k-mers in their promoters. A p-value was assigned to each EC score and false discovery rate (FDR) of 0.1 (allowing 10% false positives) was used to correct for multiple hypotheses.

4.2 Expression Coherence (EC) Score

The formal definition of the EC score is the fraction of pairs of genes in a given set S , for which the Euclidean distance between expression profiles falls below a threshold D .

$$EC(S) = \frac{\left| \{g_i, g_{j \neq i} \in S : ExpDist(g_i, g_j) < D\} \right|}{|S| * (|S| - 1) \div 2} \quad (1)$$

The threshold D is determined based on the distribution of pair-wise distances between expression profiles of all genes in the genome (or more precisely of all genes for which expression level was measured). The original definition of the EC score [13] used the 5th percentile as the cutoff for defining “close” expression profiles (D). This definition may create a bias towards TFs that exert a very tight regulation and miss regulatory motifs that correspond to factors exerting a more loose regulation. We therefore tested a range of EC definitions, with cutoffs corresponding to the 5th, 10th, 20th, 30th, 40th and 50th percentile of the pair-wise distance distribution. For each definition of EC cutoff we assigned a significance p-value separately. P-values were calculated by random sampling. For each of the 40 expression time series and for each gene set sizes (varying from 3-100 genes), we selected 100,000 random gene sets and computed an EC score for each such set at each cutoff definition. We define the p-value of a given EC score as the fraction of random sets (of the same size and condition) that scored similarly or higher (note that this sets a lower bound of 10^{-5} on the significance that can be assigned to a given EC score). Since we assume that for a given EC score, the probability to get the same score for random sets of genes drops with the set size, gene sets larger than 100 are assigned an upper bound approximated p-value, using the randomly sampled sets of size 100.

4.3 Comparing Our Binding Sites to Known PWMs

A scoring method was devised to assess how likely a given string is to be generated from a given PWM. The score is on a scale of 0 to 100. It is computed by summing up the frequencies corresponding to the observed nucleotides over all motif positions, and normalizing this score to a scale of 0-100. The scaling is done by subtracting the minimal possible score and dividing by the range of possible scores. For example for the PWM [A: 0.0191 0.0191 0.9733 0.9733 0.0120, C:0.9500 0.9500 0.0074 0.0074 0.0074, G: 0.0117 0.0117 0.0074 0.0074 0.0074 T:0.0191 0.0191 0.0120 0.0120 0.9733] the lowest possible score 0.0455 is obtained for the string GG(C/G)(C/G)(C/G), the highest possible score 4.8198 is obtained for the string CCAAT. After scaling GGCCC will score 0%, CCAAT will score 100% and CCATT will score 79.9% ($(3.8585-0.0455)/(4.8198-0.0455)$). Because our k-mers and the known PWMs may differ in length, we aligned them by sliding the shorter sequence along the longer. For each such alignment, we calculate the match score (in percentage 0-100%) and took the position with the best score as the true alignment.

References

1. Ng, P.C., Henikoff, S.: Predicting deleterious amino acid substitutions. *Genome Res* 11 (2001) 863-874
2. Ng, P.C., Henikoff, S.: Accounting for human polymorphisms predicted to affect protein function. *Genome Res* 12 (2002) 436-446

3. Ng, P.C., Henikoff, S.: SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Res* 31 (2003) 3812-3814
4. Wang, Z., Moul, J.: SNPs, protein structure, and disease. *Hum Mutat* 17 (2001) 263-270.
5. Sunyaev, S., Ramensky, V., Bork, P.: Towards a structural basis of human non-synonymous single nucleotide polymorphisms. *Trends Genet* 16 (2000) 198-200.
6. Sunyaev, S., Ramensky, V., Koch, I., Lathe, W., 3rd, Kondrashov, A.S., Bork, P.: Prediction of deleterious human alleles. *Hum Mol Genet* 10 (2001) 591-597.
7. Vitkup, D., Sander, C., Church, G.M.: The amino-acid mutational spectrum of human genetic disease. *Genome Biol* 4 (2003) R72
8. Prokunina, L., Castillejo-Lopez, C., Oberg, F., Gunnarsson, I., Berg, L., Magnusson, V., Brookes, A.J., Tentler, D., Kristjansdottir, H., Grondal, G., Bolstad, A.I., Svenungsson, E., Lundberg, I., Sturfelt, G., Jonssen, A., Truedsson, L., Lima, G., Alcocer-Varela, J., Jons-son, R., Gyllensten, U.B., Harley, J.B., Alarcon-Segovia, D., Steinsson, K., Alarcon-Riquelme, M.E.: A regulatory polymorphism in PDCD1 is associated with susceptibility to systemic lupus erythematosus in humans. *Nat Genet* 32 (2002) 666-669
9. Zwarts, K.Y., Clee, S.M., Zwinderman, A.H., Engert, J.C., Singaraja, R., Loubser, O., James, E., Roomp, K., Hudson, T.J., Jukema, J.W., Kastelein, J.J., Hayden, M.R.: ABCA1 regulatory variants influence coronary artery disease independent of effects on plasma lipid levels. *Clin Genet* 61 (2002) 115-125
10. Rockman, M.V., Wray, G.A.: Abundant raw material for cis-regulatory evolution in humans. *Mol Biol Evol* 19 (2002) 1991-2004
11. Shalgi, R., Lapidot, M., Shamir, R., Pilpel, Y.: A catalog of stability-associated sequence elements in 3' UTRs of yeast mRNAs. *Genome Biol* 6 (2005) R86
12. Harbison, C.T., Gordon, D.B., Lee, T.I., Rinaldi, N.J., Macisaac, K.D., Danford, T.W., Hannett, N.M., Tagne, J.B., Reynolds, D.B., Yoo, J., Jennings, E.G., Zeitlinger, J., Pok-holok, D.K., Kellis, M., Rolfe, P.A., Takusagawa, K.T., Lander, E.S., Gifford, D.K., Fraenkel, E., Young, R.A.: Transcriptional regulatory code of a eukaryotic genome. *Nature* 431 (2004) 99-104
13. Pilpel, Y., Sudarsanam, P., Church, G.M.: Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet* 29 (2001) 153-159
14. Sudarsanam, P., Pilpel, Y., Church, G.M.: Genome-wide Co-occurrence of Promoter Elements Reveals a cis-Regulatory Cassette of rRNA Transcription Motifs in *Saccharomyces cerevisiae*. *Genome Res* 12 (2002) 1723-1731
15. Lapidot, M., Pilpel, Y.: Comprehensive quantitative analyses of the effects of promoter sequence elements on mRNA transcription. *Nucleic Acids Res* 31 (2003) 3824-3828
16. Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. Roy Stat Soc* 57 (1995) 289-300
17. Lamoureux, J.S., Stuart, D., Tsang, R., Wu, C., Glover, J.N.: Structure of the sporulation-specific transcription factor Ndt80 bound to DNA. *Embo J* 21 (2002) 5721-5732
18. Pierce, M., Benjamin, K.R., Montano, S.P., Georgiadis, M.M., Winter, E., Vershon, A.K.: Sum1 and Ndt80 proteins compete for binding to middle sporulation element sequences that control meiotic gene expression. *Mol Cell Biol* 23 (2003) 4814-4825
19. Elkon, R., Linhart, C., Sharan, R., Shamir, R., Shiloh, Y.: Genome-wide in silico identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res* 13 (2003) 773-780
20. SGD: <http://www.yeastgenome.org/>.
21. ExpressDB: <http://salt2.med.harvard.edu/ExpressDB/>.

Comparative Systems Biology of the Sporulation Initiation Network in Prokaryotes

Michiel de Hoon and Dennis Vitkup

Columbia University, Center for Computational Biology and Bioinformatics,
New York NY 10032, United States
mdehoon@c2b2.columbia.edu

Abstract. Many years of experimental and computational molecular biology of model organisms such as *Escherichia coli* and *Saccharomyces cerevisiae* has elucidated the gene regulatory network in these organisms. Relatively little is known about gene regulation in species other than the model organisms, whether gene regulatory networks are conserved, and to what degree our knowledge based on model organisms reflects biological networks occurring in nature as a whole.

In this paper, we describe a first attempt to understand the gene regulatory network in lesser-known organisms, using our knowledge of gene regulation in a well-understood model organism. Such an extrapolation is particularly valuable in the study of disease-causing infectious agents, as well as other organisms that are difficult to grow or handle in a laboratory environment. In addition, comparative systems biology can identify which parts of biological networks are poorly understood and are therefore promising venues for further experimental research.

We analyze the gene regulatory network responsible for the initiation of sporulation in fourteen target organisms, using *Bacillus subtilis* as the model organism. Instead of focusing on individual transcription factor binding sites, we devise a scoring function that takes into account the effect of multiple transcription factors binding to the regulatory region. Whereas the core gene regulatory network appears to be conserved, the degree of conservation decreases rapidly for more remote organisms, as well as for regulatory relations in the periphery of the network. Our work shows that gene regulation is still poorly understood in species other than the model organisms.

1 Introduction

In the post-genomic era, one of the major goals of molecular biology is to understand the gene regulatory network that drives the expression of genes depending on cellular conditions. Gene regulation is mediated by transcription factors, proteins that stimulate or repress the expression of genes by binding to the regulatory DNA sequence in their upstream region. Transcription factors recognize specific motifs in the DNA sequence, enabling them to differentially regulate genes based on their respective regulatory code. In many cases, the DNA motifs recognized by a specific transcription factor are similar to each other, which

allows us to define a consensus binding motif that can be used to detect potential transcription factor binding sites in a DNA sequence.

For well-studied organisms such as *Escherichia coli* and *Saccharomyces cerevisiae*, a large number of transcription factors, their regulated genes, and the DNA binding sites have been found experimentally by DNA footprinting, disrupting transcription factors, mutating DNA binding sites, and primer extension experiments. Computationally, we can find transcription factor binding sites using comparative genomics, in which the upstream DNA sequences of homologous genes in nearby species are aligned to find conserved motifs. This approach relies on the assumption that the regulatory network is conserved between nearby species, which may or may not be true.

Whereas the combination of experimental and computational approaches has dramatically increased our knowledge of gene regulation in model organisms, relatively little is known about regulatory networks in other organisms. In this paper, we attempt to uncover the gene regulatory network in such lesser-studied organisms, using our knowledge of gene regulation in a well-studied organism.

This question can be placed in the larger context of comparative genomics. One goal of traditional comparative genomics is to infer the function of unknown proteins based on sequence homology to known proteins. More recently, the discovery of potential transcription factor binding sites by aligning gene regulatory regions across organisms has emerged as a second important goal of comparative genomics. Using sequence homology of the gene regulatory region in one organism to infer regulatory relations in another organism borrows from these two approaches, and can be seen as an emerging additional goal of comparative genomics.

To determine if the regulation of a gene in a given organism is conserved, we may attempt to search for potential binding sites of transcription factors that are known to regulate the homologous gene in a nearby organism. However, such a prediction is nontrivial for several reasons. First, since the number of experimentally known transcription factor binding sites is bounded, our statistical model of the binding motifs is necessarily limited and allows us to create only simplified models of the DNA motifs, such as position-weight matrices, or perhaps a first-order Markov model. Secondly, a transcription factor is not guaranteed to bind to each high-scoring DNA motif *in vivo*, as this may be affected by the presence of other DNA-binding proteins, the local bending of the DNA, or on cellular conditions. Third, it is often unclear if a transcription factor binding to a specific DNA site has a biological function, and if so, what the regulatory role might be. Finally, it is generally unknown if a transcription factor in a lesser-characterized organism recognizes the same DNA motifs as its homologous counterpart in a well-studied organism. Whereas aligning upstream sequences regions of homologous genes of different organisms has been successful in detecting transcription factor binding sites, it is unknown whether the conservation of binding motif extends to all transcription factors. At any rate, we expect some reduction in the accuracy of detecting transcription factor binding sites due to an imperfect conservation of the DNA binding specificity between two homologous transcription factors of different species.

While the prediction of gene regulatory networks based on a well-studied model organism is therefore a challenging task, the results that we may obtain from such an analysis are of tremendous importance. First, it allows us to assess whether our knowledge of model organisms represents biology in general, or if the gene regulatory network of each organism needs to be studied separately. Second, comparing the regulatory network in a model organism to those of lesser-known organisms will help us to identify regulatory subnetworks that do not occur in the model organisms, and that may therefore contain currently unknown biological mechanisms. Finally, it is of great medical importance to understand biological networks in disease-causing organisms. Whereas a considerable number of genome sequencing projects of such organisms have now been completed, only a fraction of their regulatory network is known. Examples of close relatives of the model organism *Bacillus subtilis*, which we consider in this paper, are *Bacillus anthracis*, which causes anthrax, *Staphylococcus aureus*, whose multiple-resistant form (MRSA) is a major source of hospital infections, and *Clostridium tetani*, the causative agent of tetanus.

Here, we focus on the regulatory network underlying initiation of sporulation in spore-forming bacteria. Gram-positive bacteria of the *Bacilli* and *Clostridia* genus have the capability of forming spores when environmental conditions become adverse. Spores, which are metabolically dormant, protect the bacterial DNA from environmental challenges such as heat, dryness, and UV radiation. As soon as the environmental conditions ameliorate, spores germinate to create the complete bacterium. Sporulation therefore helps these bacteria to survive prolonged periods of adverse environmental conditions.

The decision to sporulate has a profound effect on the survivability of a bacterium. By sporulating too soon, a bacterium inadvertently passes up additional rounds of replication, whereas failing to detect the need for sporulation may kill a bacterium altogether. *Bacillus subtilis* therefore contains an intricate regulatory network to initiate sporulation. This network connects the input of environmental sensors via two-component histidine kinases to the activation of the master regulator Spo0A, which regulates a large number of genes that are active in the initial stage of sporulation.

Transcription factors involved in sporulation as well as their regulated genes, as discovered experimentally, are collected in the DBTBS database of transcriptional regulation in *Bacillus subtilis* [1]. We use the information in this database to investigate whether the regulatory network of sporulation initiation in *Bacillus subtilis* is conserved in fourteen other fully-sequenced sporulating bacteria, ranging from the nearby *Bacillus licheniformis* to the more remote *Clostridia*.

Due to the difficulties of inferring the gene regulatory network in other organisms on the basis of sequence information, as noted above, we do not aim to identify each transcription factor binding site individually. Instead, given the combination of transcription factors that bind to the upstream region of a particular gene, we construct a joint scoring function that combines the scores of the individual binding sites. We assess the degree to which conservation is conserved

by comparing the scores obtained for the homologous genes in the fourteen target organisms to a background distribution, obtained by calculating the scores for genes known not to be involved in regulation.

2 Method

2.1 Aligning Known Transcription Factor Binding Sites

Our first task is to create a statistical model of the sequence motifs known to bind a particular transcription factor in *Bacillus subtilis*. Experimentally, transcription factor binding sites can be localized to short (about 20 nucleotide) DNA sequences, from which a conserved sequence motif (which is typically shorter) can be found by alignment. As a statistical model, we use a position-weight matrix approach [2]. The log-likelihood score of an alignment with $n(i, c)$ nucleotides c at position i can be written as

$$L = \sum_{i=1}^m \sum_{c=\{A,C,G,T\}} n(i, c) \log \frac{n(i, c)}{n}, \quad (1)$$

where m is the motif length and n is the number of sequences in the alignment; $p(i, c) = n(i, c)/n$ is the corresponding probability to find a nucleotide c at position i .

Some transcription factor binding sites consists of two sequence motifs separated by a gap for which the sequence is not conserved. In particular, the promoter sequence on the DNA, which is recognized by the σ (specificity) factor subunit of the RNA polymerase, consists of two motifs, located around 35 base-pairs and 10 base pairs upstream of the transcription start site. The two binding motifs are separated by a gap of variable length. To model such binding sites, we assume a flat probability distribution for gaps of $w_{\min}, \dots, w_{\max}$ base pairs, and write the log-likelihood as

$$\begin{aligned} L = & \sum_{i=1}^{m_{\text{left}}} \sum_{c=\{A,C,G,T\}} n_{\text{left}}(i, c) \log \left(\frac{n_{\text{left}}(i, c)}{n} \right) \\ & + \sum_{i=1}^{m_{\text{right}}} \sum_{c=\{A,C,G,T\}} n_{\text{right}}(i, c) \log \left(\frac{n_{\text{right}}(i, c)}{n} \right) \\ & - n \log (w_{\max} - w_{\min} + 1). \end{aligned} \quad (2)$$

Sigma factor recognition sites can be found experimentally by determining the starting position of the mRNA molecule in a primer extension or S1 nuclease experiment.

We aligned the known transcription factor binding sites using BioProspector [3]. As repeated runs of BioProspector returned identical alignments, we feel confident that the optimal alignment solution was found. For the sigma factor recognition sites, we used BioProspector to align the 50 basepair regions upstream of the transcription start sites, trying all plausible values of w_{\min} and

w_{\max} exhaustively and evaluating the alignment result using Equation (2). This allows us to find a statistically founded balance between the motif alignment scores and the allowable gap variation. We note that the allowable gaps we found are typically more restrictive than those identified previously [4].

To avoid the problem of overfitting, after alignment we added the pseudo-counts $q_c\sqrt{n}$, where q_c is the background probability to find a nucleotide c . Hence, the position-weight matrix for transcription factor T can be written as

$$M_T(i, c) = \log \left(\frac{(n(i, c) + q_c\sqrt{n}) / (n + \sqrt{n})}{q_c} \right) \quad (3)$$

2.2 Finding Combinations of Transcription Factor Binding Sites

Given the position-weight matrix, we can search the upstream region of genes to find likely transcription factor binding sites. To reduce the effect of false-positives, we aim to find several predicted transcription factor binding sites located within a window of length w . The score for a potential transcription factor binding site s , after Bonferoni correction for multiple comparisons over the window of length w , is calculated from the position-weight matrix as

$$\text{score}(s, \text{transcription factor } T) = \sum_{i=1}^m M_T(i, s_i) - \log(w) \quad (4)$$

The joint score that a given set of transcription factors T bind to a potential regulatory region of length w is the calculated as

$$\text{score}(\text{sequence of length } w) = \sum_T \sum_{\text{all subsequences } s \text{ of length } m_T} R(\text{score}(s, T)), \quad (5)$$

where R is the ramp function defined by

$$R(x) = \begin{cases} x & \text{if } x > 0; \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Effectively, we include every potential transcription factor binding site in the score function (5) with a positive log-likelihood score after Bonferoni correction. To determine if the regulation of a gene is conserved in a target organism, we slide a sequence window of length w along its upstream sequence region and calculate the score function (5), including only those transcription factors T in the summation that are known to regulate the gene in the source organism *Bacillus subtilis*.

To facilitate their interpretation, we normalize the joint scores as defined by Equation (5) by dividing them by the root mean square of the scores found for 2264 *Bacillus subtilis* genes that are known not to be involved in sporulation, based on their functional annotation in the SubtiList database [5,6]. As these genes are unlikely to be regulated by the sporulation-specific transcription factors, their scores can serve as a background model.

3 Results

We search the upstream DNA regions of genes whose homologs in *Bacillus subtilis* are regulated by Spo0A, the master regulator of sporulation initiation. In addition to Spo0A, these genes are regulated by various combinations of the transcription factors AbrB, AhrC, CcpA, Hpr, SinR, and SpoVT, and the sigma factors SigA, SigD, SigF, SigG, SigH, and SigX. As no binding site information is available for SpoVT, we excluded this transcription factor from our analysis. For each gene, we use Equation (5) to assess the overall resemblance of the upstream regulatory DNA sequences to their *Bacillus subtilis* counterpart.

Table 1 shows the prediction result for *Bacillus subtilis* itself, two strains of *B. licheniformis*, three strains of *B. anthracis*, three strains of *B. cereus*, one strain each of *B. thuringiensis*, *B. clausii*, *B. halodurans*, and three *Clostridia*. Gene regulation appears to be well-conserved in *Bacillus licheniformis*, for which high scores were found for nearly all genes under consideration. A lesser degree of conservation was found for the *Bacillus anthracis*, *cereus*, and *thuringiensis* strains, for which high-scoring regulatory regions were found for only half of the genes. For the *Clostridia*, regulation was usually not conserved, in particular for *Clostridium perfringens*.

We note that the degree of conservation varies strongly between genes. For the *spo0A* gene, which encodes the master regulator of sporulation initiation, regulation appears to be conserved in nearly all species, including two of the *Clostridia*. Similarly, for the *spoIIAA-spoIIAB-sigF* operon, we find a highly conserved regulatory region. Interestingly, the products of these genes play key roles in the regulation of sporulation initiation. SpoIIAB is an anti-sigma factor that inhibits SigF by binding to it; SpoIIAA is an anti-anti-sigma factor that binds to SpoIIAB, thereby releasing the inhibition of SigF. SigF is the first sporulation-specific sigma factor to be activated in sporulation, representing a major step in the initiation of sporulation. Similarly, regulation is conserved in most species for *abrB*, whose product prevents the expression of sporulation related genes until the start of sporulation, as well as for *spo0F*, a two-component response regulator involved in sporulation initiation.

Of the genes whose regulation is less conserved, *dltABCDE*, *argCJBD-carAB-argF*, and *rbsRKDACB* do not play central roles in the sporulation network. The products of the *argCJBD-carAB-argF* and *dltABCDE* operon are enzymes involved in arginine and lipothichoic acid biosynthesis, respectively; the *rbsRK-DACB* genes code for proteins involved in ribose transport and their regulator. SpoIIE, however, a serine phosphatase acting on SpoIIAA, is important in the sporulation initiation network. The regulatory region of *spoIIE* consists of several weak Spo0A binding sites, none of which are strong enough to overcome the Bonferoni correction. Hence, we are unable to detect the regulatory region even in *Bacillus subtilis* itself. Similarly, for the *spoIIGA-sigE-sigG* operon, also important for the initiation of regulation, only one of the weak transcription factor binding sites can be detected in *Bacillus subtilis*.

For *kinC* and *kinA*, whose products act as two-component sensor histidine kinases in the phosphorylation pathway of sporulation initiation, the main

Table 1. Conservation of regulation of sporulation initiation in *Bacilli* and *Clostridia*. The listed numbers are the scores calculated from Equation (5), divided by the root mean square of the scores calculated for genes known not to be involved in regulation. To calculate the score, we include those transcription factors that are known to regulate the gene in *Bacillus subtilis*. A dash indicates that no orthologous gene could be identified in the genome.

Organism	<i>abrB</i>	<i>argC</i> ^a	<i>dltA</i> ^b	<i>kinA</i>	<i>kinC</i>	<i>rbsR</i> ^c	<i>sinI</i> ^d	<i>spo0A</i>	<i>spo0F</i>	<i>spoIIA</i> ^e	<i>spoIII</i>	<i>spoIIIGA</i> ^f
<i>B. subtilis</i>	5.1	2.9	8.2	5.0	0.2	2.7	1.8	4.1	3.0	4.0	0.0	0.4
<i>B. licheniformis</i> Novozymes Biotech	2.3	2.4	5.1	5.3	0.0	0.9	0.1	3.8	2.3	4.6	0.0	0.4
<i>B. licheniformis</i> Göttingen	2.3	2.4	5.1	5.3	0.0	0.2	0.1	3.8	2.3	4.6	0.0	0.4
<i>B. anthracis</i> str. ‘Ames Ancestor’	2.3	2.0	0.0	0.0	0.3	0.2	1.9	4.8	5.7	4.0	0.0	0.0
<i>B. anthracis</i> str. ‘Ames’	2.3	2.0	0.0	0.0	0.3	0.2	1.9	4.8	5.7	4.0	0.0	0.0
<i>B. anthracis</i> str. Sterne	2.3	2.0	0.0	1.5	0.3	0.2	1.9	3.5	5.7	4.0	0.0	0.0
<i>B. cereus</i> ATCC 10987	2.3	2.0	0.0	0.0	3.1	0.2	1.9	3.7	4.4	3.7	0.0	0.0
<i>B. cereus</i> ATCC 14579	2.3	0.3	0.0	0.7	0.0	0.2	0.0	4.8	5.6	4.0	0.0	0.0
<i>B. cereus</i> ZK	1.2	2.0	0.0	0.0	1.4	0.0	-	3.5	5.2	4.0	0.0	0.1
<i>B. thuringiensis</i>	0.0	2.0	0.0	0.0	0.9	0.4	-	3.5	5.8	4.0	0.0	0.0
<i>B. clausii</i>	5.5	1.2	-	0.0	0.0	0.0	-	3.8	3.3	2.2	2.8	0.8
<i>B. halodurans</i>	6.5	0.5	-	0.0	0.8	1.0	0.0	3.4	4.2	1.7	0.0	0.4
<i>C. acetobutylicum</i>	0.0	2.4	-	1.0	0.0	0.5	-	4.0	-	3.5	0.2	0.0
<i>C. perfringens</i>	4.1	-	-	0.0	1.6	0.5	-	1.4	-	0.1	0.0	1.5
<i>C. tetani</i>	0.0	-	-	2.4	0.0	1.1	-	4.3	-	2.4	0.0	0.0

^a *argC*/*JBD-carAB-argF* operon

^b *dltABCDE* operon

^c *rbsRKDACB* operon

^d *sinIR* operon

^e *spoIIA-spoIIAB-sigF* operon

^f *spoIIGA-sigE-sigG* operon

difficulty lies in the determination of their homologs in other species. At least five histidine kinases, with overlapping roles, participate in the initiation of sporulation in *Bacillus subtilis*. As no clear one-to-one relation exists between histidine kinases in different organisms, except in the most nearby organisms, we do not expect to be able to find a strong conservation of their regulatory regions.

4 Discussion

The example of the regulation of sporulation initiation demonstrates some of the potentials and pitfalls of comparative systems biology.

Our analysis of the regulatory regions of genes involved in sporulation initiation reveals that the core of the network appears to be conserved between organisms, whereas more peripheral parts of the network are poorly conserved. In addition to the difficulty in recognizing regulatory regions, the identification of orthologous genes in different can be non-trivial. We found this especially to be the case for the histidine kinases, which are important in phosphorylation signalling pathways in the initiation of sporulation. This suggests that a comprehensive approach, in which signalling pathways and gene regulatory relations are reconstructed simultaneously, may give a clearer view of the conservation of regulatory networks.

As our scoring function (Equation (5)) takes into account the contribution of several transcription factors, it is more powerful in detecting potential regulatory regions and their conservation. However, a more detailed analysis of regulatory DNA, in which individual transcription factor binding sites are identified (if feasible) is preferable. We feel that currently, our ability to detect transcription factor binding sites is not sufficiently powerful for such a prediction. In addition to improving the accuracy of predicting transcription factor binding sites, another challenge is to determine which of the transcription factor binding sites fulfill a biological function, and if so, what that function is.

References

1. Makita, Y., Nakao, M., Ogasawara, N., and Nakai, K.: DBTBS: Database of transcriptional regulation in *Bacillus subtilis* and its contribution to comparative genomics. *Nucleic Acids Res.*, 32 (2004) D75–77.
2. Durbin, R., Eddy, S.R., Krogh, A., and Mitchison, G.: *Biological sequence analysis*. Cambridge University Press, Cambridge, UK (1998).
3. Liu, X., Brutlag, D.L., and Liu, J.S.: BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, 6 (2001) 127–138.
4. Sonenshein, A.L., Hoch, J.A., and Losick, R.: *Bacillus subtilis* and its closest relatives: from genes to cells. ASM Press, Washington, DC (2001).
5. Moszer, I., Glaser, P., and Danchin, A.: SubtiList: a relational database for the *Bacillus subtilis* genome. *Microbiology*, 141 (1995) 261–268.
6. Moszer, I.: The complete genome of *Bacillus subtilis*: From sequence annotation to data management and analysis. *FEBS Letters* 430 (1998) 28–36.

Improvement of Computing Times in Boolean Networks Using Chi-square Tests

Haseong Kim¹, Jae K. Lee², and Taesung Park³

¹ Interdisciplinary Program in Bioinformatics, Seoul National University,
Seoul, Korea

khs@bibs.snu.ac.kr

² Division of Biostatistics and Epidemiology, University of Virginia,
Charlottesville, USA

jaeklee@virginia.edu

³ Department of Statistics, Seoul National University,
Seoul, Korea

tspark@snu.ac.kr

Abstract. Boolean network is one of the commonly used methods for building gene regulatory networks from time series microarray data. However, it has a major drawback that requires heavy computing times to infer large scale gene networks. This paper proposes a variable selection method to reduce Boolean network computing times using the chi-square statistics for testing independence in two way contingency tables. We compare the computing times and the accuracy of the estimated network structure by the proposed method with those of the original Boolean network method. For the comparative studies, we use simulated data and a real yeast cell-cycle gene expression data (Spellman *et al.*, 1998). The comparative results show that the proposed variable selection method improves the computing time of Boolean network algorithm. We expect the proposed variable selection method to be more efficient for the large scale gene regulatory network studies.

1 Introduction

1.1 Boolean Networks

Boolean network models were first introduced by Kauffman (1969). In Boolean network models, gene expression is quantized to only two levels: ON and OFF. A Boolean network $G(V, F)$ is defined by a set of nodes $V = \{x_1, \dots, x_n\}$ and a list of Boolean functions $F = \{f_1, \dots, f_n\}$. A Boolean function, $f_i(x_1, \dots, x_k)$, $i = \{1, \dots, n\}$ with k specified input nodes (indegree) is assigned to node x_i . Regulation of nodes is defined by set F of Boolean functions. In detail, given the value of the nodes V at time t , the Boolean functions are used to update the value of the nodes at time $t+1$.

The model system was developed into so-called random Boolean network model (Kauffman, 1993). Recently, Boolean networks have attracted a public attention, since the probabilistic Boolean network models were introduced by Shmulevich *et al.* (2002a). Many algorithms have been proposed for inference of Boolean networks. For example, REVEL algorithm was introduced by Liang *et al.* (1998) for the causal inference using the mutual information that is the most fundamental and general

correlation measure. Akutsu *et al.* (1999) built the Boolean network structure using so-called Consistency Problem, by which one can determine whether there exists a network consistent with the observed data. One of the most recent works of Boolean network algorithm was conducted by Shmulevich *et al.* (2002b), in which Best-Fit Extension of Boros *et al.* (1998) was used for the inference of Boolean networks.

Recently, several software packages have been developed for constructing Boolean networks. The Random Boolean network toolbox (Schwarzer, 2003) and probabilistic Boolean network toolbox (Shmulevich *et al.*, 2002b) is available in Matlab. Netbuilder (version 0.94) (Schilstra *et al.*, 2003) is a genetic regulatory network tool to simulate the genetic network using Boolean network.

The Boolean network has been widely used to describe the biological process. For example, cell growth, cell differentiation, and apoptosis were represented by Huang (1999). The transcriptional network model in yeast was studied using Random Boolean network (Kauffman, 2003). Johnson (2004) studied the signal transduction pathways in B-cell ligand screen.

1.2 Advantages of Boolean Networks

There are several advantages to construct gene regulatory network using Boolean networks. First, Boolean network model explains the dynamic behavior of living systems efficiently. Simplistic Boolean formalism can represent the realistic complex biological phenomena such as cellular state dynamics, possessing switch-like behavior, stability, and hysteresis (Huang, 1999). Second, Boolean algebra is a mature science, providing a rich set of algorithms already available for supervised learning in binary domain, such as logical analysis of data (Boros *et al.*, 1997), and Boolean-based classification algorithms (Akutsu *et al.*, 2001). Finally, dichotomization to binary values improves accuracy of classification and simplifies the obtained models by reducing the noise level in experimental data (Pfahring, 1995; Dougherty *et al.*, 1995).

1.3 Drawback of Boolean Networks

On the other hand, Boolean network has some drawbacks. One of the major drawbacks is that it requires heavy computing times to construct a network structure. Most Boolean network algorithms such as REVEAL can only be used with a small number of genes and a low indegree value. For higher indegree values, these algorithms should be accelerated through parallelization and increase the search efficiency of solution space (Liang *et al.*, 1998). For a restricted class of Boolean functions, Consistency Problem works in $O(2^{2^k} \cdot {}_n C_k \cdot m \cdot n \cdot \text{poly}(k))$ time (m is observed time points, n is total number of genes and $\text{poly}(k)$ means the time to compare one pair of examples), for fixed indegree k (Akutsu *et al.*, 1999) because there are a total of 2^{2^k} Boolean functions that must be checked for each of the ${}_n C_k$ possible combinations of variables and for m observations. Best-Fit Extension algorithm has $O(2^{2^k} \cdot {}_n C_k \cdot n \cdot m \cdot \text{poly}(k))$ time complexity (Shmulevich *et al.*, 2002b). Although the improved Consistency algorithm and Best-Fit Extension algorithm works in $O({}_n C_k \cdot n \cdot m \cdot \text{poly}(k))$ (Lähdesmaki *et al.*, 2003), these methods still have exponential increase of computing time for parameters n and k . This heavy computing time is a

major restriction to study large scale gene regulatory and interaction systems by Boolean network. In this paper, we use a Chi-square testing (CST)-based variable selection method together with Best-Fit Extension algorithm with three Boolean operator, AND, OR and NOT, in order to find the most relevant Boolean functions efficiently.

The reduced time complexity is $O(2^{2k-1} \cdot \sum_{i=1}^n n_i C_k \cdot m \cdot \text{poly}(k))$ where n_i is selected number of genes using independence test for searching i th Boolean function.

1.4 Independence Test in Two-Way Contingency Table

Dichotomization of the continuous gene expression values produces a two-way contingency table and allows us to perform the independence test. We use the chi-square test to identify genes that are related with a target gene. A target gene would be expressed in accordance with a Boolean function having indegrees related to the selected genes. Since the genes have only two levels $\{0, 1\}$, we use 2×2 contingency tables to identify the relationship between the two genes. The proposed method is described in detail in the next section.

2 Method

2.1 Variable Selection Using the Chi-square Test in Two-Way Contingency Table

Let n be the total number of genes. At the first step of analysis, we need to make 2×2 contingency tables from the dichotomized gene expression data. The columns of table consist of the i th gene expression level at time t , and the rows the j th gene expression level at time $t-1$ ($i=1, \dots, n$, $j=1, \dots, n$). For these i th and j th genes, we can construct a 2×2 contingency table with four cells. Each cell has a frequency of $\{0,0\}, \{0,1\}, \{1,0\}$ and $\{1,1\}$, where $\{0,0\}$ represents the i th gene level at time t is 0 and j th gene level at time $t-1$ is 0, and so forth.

The second step of variable selection method calculates a chi-square test statistic for testing the independence between two genes. For multinomial sampling with probabilities $\{\pi_{pq}\}$ in a 2×2 contingency table where the null hypothesis of independence is $H_0 : \pi_{pq} = \pi_{p+} \pi_{+q}$ (i th gene at time t and j th gene at time $t-1$ are independent) for all p and q . The usual Pearson's chi-square test can be used to test H_0 using the observed frequency, O_{pq} , and the expected frequency, E_{pq} , under H_0 . For the continuity correction, we add an arbitrary small number a to the each observed frequency to prevent some E_{pq} from being zero (Agresti, 1994). We use the value of a 0.1. Usually, $\{\pi_{p+}\}$ and $\{\pi_{+q}\}$ are unknown. Their maximum likelihood (ML) estimates are the sample marginal proportions $\hat{\pi}_{p+} = O_{p+} / O$ and $\hat{\pi}_{+q} = O_{+q} / O$, so the estimated expected frequencies are $\{E_{pq} = O \hat{\pi}_{p+} \hat{\pi}_{+q} = O_{p+} O_{+q} / O\}$. Thus, the chi-square statistic is given by

$$X_{ij}^2 = \sum_p \sum_q \frac{(O_{pq} - E_{pq})^2}{E_{pq}} \quad (1)$$

The last step of the proposed method is selection of the significant genes using an appropriate selection criterion. Instead of using the usual significant level $\alpha=0.05$, we use a fixed criterion c which value is much larger than 0.05. In the result section, the error rate is used to get a proper value of c . Testing the independence about all possible $\{i, j\}$ pairs of genes would select n_i genes ($0 < n_i < n$) for the i th gene. The Boolean network algorithm using the proposed method has $O(2^{2k-1} \cdot n_i C_k \cdot m \cdot \text{poly}(k))$ time complexity to find a Boolean function for the i th gene. Thus, the total time complexity of the proposed algorithm is given by

$$O(2^{2k-1} \cdot \sum_{i=1}^n n_i C_k \cdot m \cdot \text{poly}(k)) \quad (2)$$

where $0 < n_i < n$. If $n_i < k$ then $n_i C_k = n_i$. We do not consider about the time complexity of variable selection, $O(n^2)$, because it has a small additive effect to the overall time complexity.

Example 1. Consider a simple network structure consisting of four nodes $V=(G1, G2, G3, G4)$ and the functions for each nodes $F=(f_1, f_2, f_3, f_4)$, are shown in Fig. 1. Table 1 shows the data matrix with 10 time points from the given network structure.

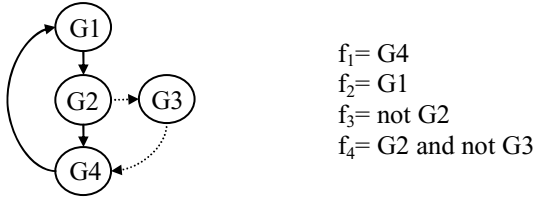


Fig. 1. A diagram of a simple network example and its Boolean functions. Arrowed lines represent activation and dotted arrow lines represent inhibition.

Table 1. Binary data set from simple network in Fig. 1. Total of four nodes (G1, G2, G3, G4) and 10 time points. 0 represent that the gene is unexpressed and 1 represent that the gene is expressed.

	G1	G2	G3	G4
T1	0	1	0	1
T2	1	0	0	1
T3	1	1	1	0
T4	0	1	0	0
T5	0	0	0	1
T6	1	0	1	0
T7	0	1	1	0
T8	0	0	0	0
T9	0	0	1	0
T10	0	0	1	0

If we want to find a Boolean function, f_4 for node G_4 , we have to test the independence between G_4 at time t and other nodes (G_1, G_2, G_3, G_5) at time $t-1$. Four 2×2 contingency tables are constructed for these tests. Fig. 2 shows two-way contingency tables for the independence tests between G_4 and the other nodes. (a), (b), (c), and (d) represent four pair wise contingency tables, ($G_{4_t}, G_{1_{t-1}}$), ($G_{4_t}, G_{2_{t-1}}$), ($G_{4_t}, G_{3_{t-1}}$), and ($G_{4_t}, G_{4_{t-1}}$) respectively.

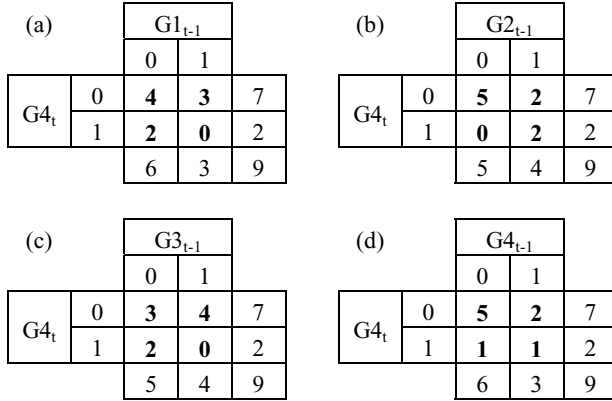


Fig. 2. Contingency tables derived from the data matrix in Table 1

Chi-square statistics have chi-square distribution with 1 degree of freedom. Table 2 shows the result of independence test for G_{4_t} node. It shows that the G_4 node at time t has a relationship with G_1, G_2 and G_3 at time $t-1$ when $c=0.5$ (gray color). If we do not use the proposed variable selection method to find Boolean functions, f_4 , we have to search all possible combinations, ${}_4C_2=6$ (when indegree $k=2$). Using the proposed variable selection method, we can find a function, $f_4 = G_2$ and not G_3 , which can also be obtained by the original Boolean network at a time (${}_3C_2=3$).

Table 2. Result of independence test for G_{4_t} node

	G1 _{t-1}	G2 _{t-1}	G3 _{t-1}	G4 _{t-1}
X ²	1.113	2.995	2.037	0.325
p-value	0.291	0.083	0.153	0.568

3 Result

3.1 Simulation Data Set

We construct an artificially generated network structure (Fig. 3). The network structure consists of eight nodes and the maximum value of indegree k is 2. We obtain four set of binary data without noise from the network structure. Each data set has different initial states and ten time points. Table 3 shows the p-values of the chi-square test between the i th gene at time t and the j th gene at time $t-1$ when c is 0.01. The selected

variables (gray color) having significant p-values are the candidates of essential variables in each Boolean function (Fig. 3 (b)).

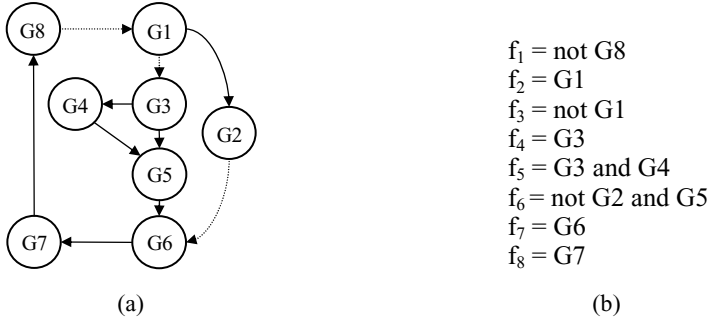


Fig. 3. (a) Artificially generated network. (b) The Boolean functions of each node in the network structure.

Table 3. *p*-values of independence test for each pair of nodes in simulation data

$i \backslash j$	G1	G2	G3	G4	G5	G6	G7	G8
G1	0.017	0.298	0.298	0.087	0.025	0.237	0.009	2e-9
G2	2e-9	0.048	0.048	0.517	0.139	0.060	0.273	0.017
G3	2e-9	0.048	0.048	0.517	0.139	0.060	0.273	0.017
G4	0.048	3e-6	2e-9	0.189	0.139	0.241	0.075	0.298
G5	0.060	0.001	6e-5	6e-5	0.000	0.134	0.092	0.237
G6	0.086	0.001	0.013	0.013	4e-6	0.014	0.237	0.440
G7	0.060	0.241	0.241	0.060	0.000	2e-9	0.016	0.237
G8	0.273	0.075	0.075	0.273	0.037	0.016	2e-9	0.009

The Boolean network algorithm using the proposed variable selection method estimated the Boolean functions which are the exactly same with the result of original Boolean network algorithm. However, there is a big difference between computing times between two algorithms. We could get a ratio of the time complexities of the two methods.

$$\frac{\text{Time complexity of Boolean networks with variable selection}}{\text{Time complexity of original Boolean networks}}$$

$$\begin{aligned}
 & \frac{O(2^{2k-1} \cdot \sum_{i=1}^n n_i C_k \cdot m \cdot \text{poly}(k))}{O(2^{2k-1} \cdot n C_k \cdot n \cdot m \cdot \text{poly}(k))} \\
 &= \frac{\sum_{i=1}^n n_i C_k}{n C_k \cdot n} = \frac{(2 C_2 + 1 + 1 + 2 C_2 + 4 C_2 + 2 C_2 + 2 C_2 + 2 C_2)}{8 C_2 \cdot 8} \\
 &= 0.058
 \end{aligned}$$

(3)

As shown in Equation (3), the Boolean networks with the proposed variable selection algorithm is about 20 times faster than original Boolean networks in this simulation study. If the network has larger nodes (n) and indegree (k), the difference of computing times between two algorithms would increase exponentially.

3.2 Yeast Cell Cycle Data

In order to illustrate a more explicit improvement of computing times, we apply the proposed variable selection method to yeast cell-cycle data (Spellman *et al.*, 1998). It has 28 time points (alpha factor based synchronization experiment). In this example, we focus on the comparison of computing times and the accuracy of estimated networks between the original Boolean network and the Boolean network based on the proposed variable selection method. We did not attempt to validate the estimated network structure biologically using our Boolean network algorithm.

3.2.1 Comparison of Network Structure Estimation Accuracy Between the Boolean Network Algorithm with Variable Selection Method and Original Boolean Networks

Our variable selection method greatly improves the computing time of Boolean networks. However, the accuracy of our method should also be assessed before comparing the computing times between two methods. The improvement of computing times largely depends on the value c which is used to select genes at time $t-1$ which is related with a gene at time t using the chi-square test between them. Depending on the choice of appropriate value of c , the Boolean network with variable selection method may not find some Boolean functions that may be found by using original Boolean network algorithm. It may be caused by the missing essential variables using too stringent value of c .

In this section, we show the error rate which is defined as a discrepancy between the estimated Boolean functions from the original Boolean network algorithm and that using the proposed variable selection method. In Fig. 4, the y -axis is the error rate and the x -axis is p -values. As the p -value increases, the error rate decreases, which implies that a less conservative p -value criterion would be desirable to construct a Boolean

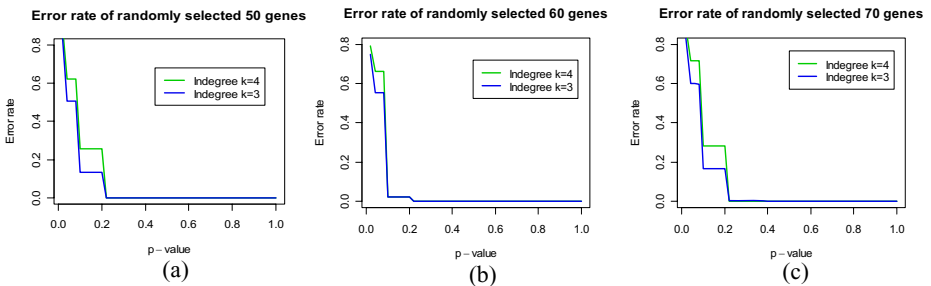


Fig. 4. Error rate of estimated Boolean functions using the chi-square of independence test for the randomly selected 50(a), 60(b), and 70(c) genes

network. We computed the error rate test with three data sets having randomly selected 50, 60, and 70 genes, respectively (Fig. 4. (a), (b), (c)). The blue and green lines represent the cases when the values of indegree k are 3 and 4, respectively. The error rate decreases, as the p -value increases. Thus, as the value of criteria, c , decreases, the computing time reduces, and also the accuracy decreases. The error rates are zero when the p -values are higher than 0.22(a), 0.46(b), and 0.38(c). Based on these results, the selection criterion with $c=0.5$ is adopted in this study.

3.2.2 Comparison of Computing Times Between the Original Boolean Network Algorithm and the Proposed Boolean Network Algorithm

To compare the computing times, we ran the Boolean network program written by C language based on Best-Fit problem (Shmulevich *et al.*, 2002b). Fig. 5 shows the computing times with varying number of genes from 40 to 120. We set the value of indegree $k=3$ and $k=4$ in Boolean network program (Fig. 5. (a) and (b) respectively) and used the variable selection criteria $c=0.5$. Blue line represents the computing time of the original Boolean network and the green line does that of Boolean network method using the proposed variable selection method. In Fig. 5 (a) original Boolean networks took 5,030 seconds to estimate all Boolean functions with 120 genes for $k=3$. On the other hand, Boolean network with the proposed variable selection method took only 1310 seconds under the same condition. When $k=4$, more improved computing times were obtained (Fig. 5 (b)). The original Boolean network method took 1,125,360 seconds to find Boolean functions but the proposed Boolean network with variables selection method took only 148,295 seconds for this case. Thus, the computing times of the Boolean network with the proposed variables selection is about 7.6 times faster than that of the original Boolean network method.

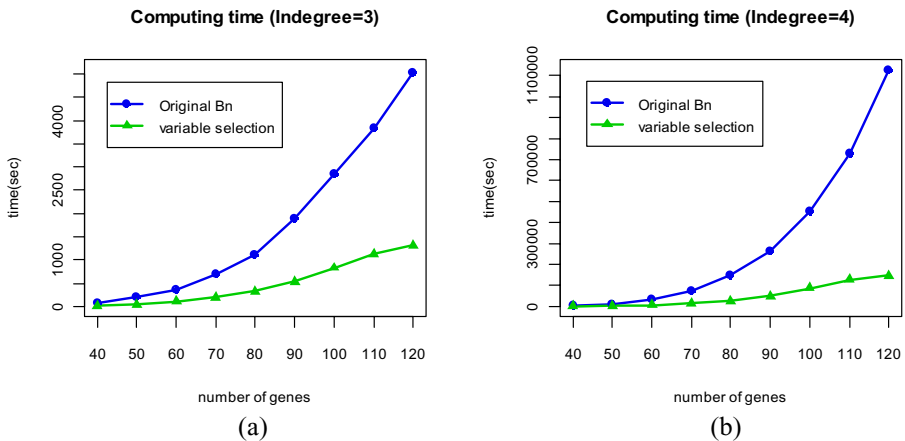


Fig. 5. Result of computing times when the total number of genes varies from 40 to 120 for $k=3$ (a) and $k=4$ (b). The line with triangles represents the computing times of the proposed Boolean network and the line with dots represent the computing times of original Boolean networks.

4 Discussion

Boolean network method is useful in building a gene regulatory network. If the gene expression data contain a considerable amount of noise, the binary transformation of the expression data can significantly reduce the error (Shmulevich and Zang, 2002). Despite of this advantage, the Boolean network method is difficult to apply to large scale gene regulatory network studies due to the heavy computing times. In order to lessen this computing burden, we propose a variable selection method using the chi-square test on the two-way contingency tables of Boolean count observations. The proposed variable selection method reduces the computing times significantly; for example, the computing time with total 120 genes of our proposed algorithm is about 7.6 times faster than previous work. If the total number of genes and indegree value k increase, we expect the proposed method improve the computing time exponentially compared to the original Boolean network method. Moreover, the proposed method is easy to implement in Boolean network model constructions.

In order to apply the proposed variable selection method, we first need to choose the value of c . A small value of c causes the exclusion of essential Boolean functions and a high error rate. On the other hand, a large value of c including most of genes may results in heavy computing times. Although this limitation, our empirical study shows the noticeable improvement of computing time even with a relatively large value of criteria c .

In addition, a more careful binary transformation may be required for the biological interpretation of the network structure. For example, since the microarray data have continuous expression values with much richer information, dichotomization may require the choice of an appropriate threshold value that is relevant to each gene's biological function (Shmulevich and Zang, 2002). The performance may not be dramatic when small numbers of time points and genes are considered in the Boolean network modeling, but we have shown that the proposed variable selection method is significantly more efficient for the large scale gene regulatory network studies.

Acknowledgements

The work was supported by the National Research Laboratory Program of Korea Science and Engineering foundation.

References

1. Agresti A.: Categorical data analysis, second edition. Wiley-interscience. (2002)
2. Akutsu T., Miyano S., Kuhara S.: Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. Pacific Symposium on Bio-computing (1999) 4:17-28
3. Akutsu T., Miyano S.: Selecting informative genes for cancer classification using gene expression data. In Proceedings of the IEEE-EURASIP Workshop on NonlinSignal and Image Processing (NSIP). Baltimore, MD (2001) 3-6

4. Boros E., Hammer P.L., Ibaraki T., and Kogan A.: Logical analysis of numerical data. *Math. Program* (1997) 79:163-190
5. Boros E., Ibaraki T., Makino K.: Error-Free and Best-Fit Extensions of partially defined Boolean functions. *Information and Computation*. (1998) 140:254-283
6. Dougherty J., Kohavi R., and Sahami M.: Supervised and unsupervised discretization of continuous features. In *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, Tahoe City, CA. (1995) 194-202
7. Huang S.: Gene expression profiling, genetic networks and cellular states: An integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine* (1999) 77:469-480
8. Johnson, S.: Boolean network inference and experiment design for the B-Cell single ligand screen, AfCS annual meeting (2004)
9. Kauffman, S., Peterson, C., Samuelsson, B., Troein, C.: Random boolean network models and the yeast transcriptional network. *Proc. Natl Acad. Sci. USA*, 100 (2003) 14796-14799
10. Kauffman SA.: Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* (1969) 22:437-467
11. Kauffman SA.: *The Origins of Order: Self-organization and Selection in Evolution*. New York:Oxford University Press (1993)
12. Lähdesmaki H., Shmulevich I., Yli-Harja O.: On learning gene regulatory networks under the Boolean network model. *Machine Learning* (2003) 52 147-167
13. Liang S., Fuhrman S., Somogyi R.: REVEAL, A general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing* (1998) 3:18-29
14. Pfahringer B.: Compression-based discretization of continuous attributes. In *Prieditis, A. and Russell, S. (eds), Machine Learning: Procees of the Twelfth International Conference*. Morgan Kaufmann, San Francisco (1995)
15. Schilstra, M. J., Bolouri, H.: Modeling the regulation of gene expression in genetic regulatory networks (2003) <http://strc.herts.ac.uk/bio/aria/NetBuilder/>
16. Schwarzer, C.: Matlab Random Boolean Network Toolbox (2003) <http://www.teuscher.ch/rbntoolbox/index.html>
17. Shmulevich I., Dougherty ER., Seungchan K., Zhang W.: Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* (2002a) 18 261-274
18. Shmulevich I., Saarinen A., Yli-Harja O., Astola J.: Inference of genetic regulatory networks under the Best-Fit Extension paradigm. In *Computational And Statistical Approaches To Genomics*, W.Zhang, and I. Shmulevich(Eds.), Boston: Kluwer Academic Publishers (2002b).
19. Shmulevich I., Zang W.: Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics* (2002) vol. 18 no. 4 555-565
20. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*. (1998) 9 3273-3297.

Build a Dictionary, Learn a Grammar, Decipher Stegoscripts, and Discover Genomic Regulatory Elements

Guandong Wang¹ and Weixiong Zhang^{1,2,*}

¹ Department of Computer Science and Engineering,

² Department of Genetics

Washington University in Saint Louis

Saint Louis, MO 63130-4899, USA

{gw2, zhang}@cse.wustl.edu

Abstract. It has been a challenge to discover transcription factor (TF) binding motifs (TFBMs), which are short *cis*-regulatory DNA sequences playing essential roles in transcriptional regulation. We approach the problem of discovering TFBMs from a steganographic perspective. We view the regulatory regions of a genome as if they constituted a stegoscript with conserved words (i.e., TFBMs) being embedded in a cocontext, and model the stegoscript with a statistical model consisting of a dictionary and a grammar. We develop an efficient algorithm, *WordSpy*, to learn such a model from a stegoscript and to recover conserved motifs. Subsequently, we select biologically meaningful motifs based on a motif's specificity to the set of genes of interest and/or the expression coherence of the genes whose promoters contain the motif. From the promoters of 645 distinct cell-cycle related genes of *S. cerevisiae*, our method is able to identify all known cell-cycle related TFBMs among its top ranking motifs. Our method can also be directly applied to discriminative motif finding. By utilizing the ChIP-chip data of Lee *et al.*, we predicted potential binding motifs of 113 known transcription factors of budding yeast.

1 Introduction

Gene expressions are mainly regulated at the transcriptional level by the binding of transcription factors (TFs) and short *cis*-acting DNA motifs [1]. Discovering TF binding motifs (TFBMs) is of fundamental importance for elucidating transcriptional regulation mechanisms.

Over the past several years, the development of high-throughput gene expression profiling technologies and the availability of a large number of complete genome sequences have made it practical to computationally discover potential TFBMs. Many motif finding algorithms, including multiple local alignment based [2,3,4,5] and word enumeration based [6,7], have been developed to find several significant motifs in the upstream regions of a small group of genes. Most existing motif finding algorithms are effective on small sets of sequences. A widely adopted method for finding such small

* Corresponding author. Phone: (314)935-8788; Fax: (314)935-7302.

sets of genes is to cluster the genes of interest so that the ones in a cluster have similar expression patterns over different conditions. This common practice is based on the assumption that co-expressed genes are co-regulated.

However, finding and using co-expressed genes to infer co-regulation, so as to identify common TFBMs, may not be effective or may even be problematic in some cases. Computational gene clustering is inaccurate and even subjective, in terms of what similarity measure to use and how many clusters to form. Importantly, many genes belonging to a common pathway may have similar expression patterns, but are not regulated by the same TFs. Therefore, co-expression does not necessarily mean co-regulation. Furthermore, gene regulation is combinatorial [1], in that one TFBM may combine with various other TFBMs to function under different conditions. This means that a TFBM may appear in the promoters of genes expressed differently. Therefore, clustering genes into small sets based on their expression patterns may split the genes of the motif into different clusters, which makes it difficult, if not impossible, to find all TFBMs [8].

With the growing importance of system biology in the post genomic era, promoter and motif analysis becomes an imperative step in understanding gene regulation in a genomic scale. For instance, a comprehensive motif analysis is critical in constructing gene regulatory networks [9,10]. This requires identifying all motifs in a genome scale, from the whole set of promoter sequences of the genes involved in a complex process.

In this paper, we propose a genome-wide motif finding approach. We first find all over-represented motifs from the promoters of the genes of interest, for instances, all genes related to a particular biological process such as cell cycle, or the genes responsive to specific physiological conditions such as cancer. We then use gene expressions and/or other information to evaluate the biological relevance of the motifs, so as to find true TFBMs. Compared to the conventional method of clustering genes first followed by motif finding, our approach finds motifs first and then selects the relevant ones.

We approach the problem of genome-wide motif finding from the perspectives of steganography and steganalysis. Steganography is a technique for concealing the existence of information by embedding the messages to be protected in a cocontext to create a stegoscript [11]. Steganalysis is just the opposite; it is to decipher a stegoscript by discovering the hidden messages [11]. We consider the regulatory regions of a genome as though they constituted a stegoscript with over-represented words, i.e., TFBMs, embedded in a cocontext. We then model the stegoscript with a statistical model consisting of a dictionary and a grammar. To decipher the stegoscript, we progressively learn a series of models that most likely generated the script. Based on this novel viewpoint, we develop an efficient genome-wide motif finding algorithm, called *WordSpy*, that can simultaneously discover a large number of motifs from a large collection of sequences. Note that our technical approach of using dictionary is inspired by the work of Bussemaker *et al.* [12], in which they introduced the innovative concepts of segmenting the sequences into words and building a dictionary of over-represented words from the sequences. Furthermore, we augment the basic algorithm by incorporating gene expression information and a genome-wide Monte Carlo simulation to distinguish biologically relevant motifs from spurious ones. We evaluate the method with an English stegoscript and 645 distinct cell-cycle related genes of *S. cerevisiae*. We also apply *WordSpy* as a

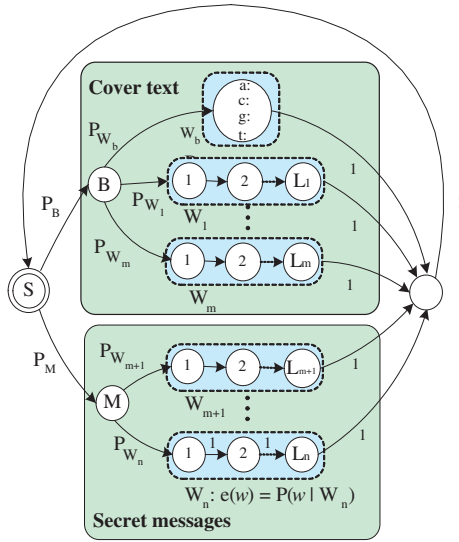


Fig. 1. A hidden Markov model for deciphering stegoscripts. It consists of two submodels, the *secret message model* is for motifs and the *covertext model* for background words. A dash box represents a word node, which is a combination of several position nodes. Node W_b is a single-base node and always belongs to the covertext model. States S , B and M do not emit any letter.

discriminative motif finding algorithm by incorporating TF location information, e.g., ChIP-chip data, and build a dictionary of motifs for each known TF of budding yeast.

2 Methods

2.1 Stegoscripts and Statistical Model

Transcriptional regulation is combinatorial; a small number of TFs mediate a large number of genes, making TFBMs over-represented in a genome. Despite their over-representativeness, TFBMs are hard to identify precisely, because, in contrast to the coding regions of a genome, our knowledge of the intergenic regions is limited. Numerous repeats or fake motifs are also dispersed in the intergenic regions, making it difficult to discern biologically meaningful motifs from spurious ones.

Regulatory genomic regions typically consist of conserved, short regulatory elements as well as a large amount of “random” sequences with no known function. We can thus view regulatory sequences as a stegoscript, and subsequently model it statistically. The model captures degenerate TFBMs and background words by a dictionary, and specifies how the motifs and words are used to form the stegoscript by a grammar [13]. Here, a grammar for a formal language is a set of rules by which all possible strings in the language can be generated by successively rewriting strings starting from a designated start symbol.

Fig. 1 illustrates our model. It can be used to generate a stegoscript as well as decipher it. At each step of generating a script, a motif M is chosen with probability P_M , or

a background word B is selected with probability P_B . Once M is chosen, a degenerate motif W_i is drawn, with probability P_{W_i} , from the *motif subdictionary*, and an exact word w is generated with probability $P(w|W_i)$. Similar is the process for the background word B . The generated word is then appended to the script created so far and the process repeats until the whole script is created.

The key to deciphering a stegoscript or a set of regulatory sequences is to learn a statistical model from which the script was supposedly created. Assume that a stegoscript \mathcal{S} were generated from an unknown model $\langle D^*, G^* \rangle$ with a dictionary D^* and a grammar G^* . We are particularly interested in such a model $\langle D, G \rangle = \arg \max_{\langle D', G' \rangle} P(\langle D', G' \rangle | \mathcal{S})$ that most likely generates the script. With no prior knowledge of the true model, the maximum likelihood estimation, $\langle D, G \rangle = \arg \max_{\langle D', G' \rangle} P(\mathcal{S} | \langle D', G' \rangle)$, is a good approximation of $\langle D^*, G^* \rangle$. In order to make the learning computationally feasible, we assume that individual motifs and words were used independently. We thus use a stochastic regular grammar [13], which is equivalent to a hidden Markov model (HMM) [14], in our statistical model. As we will see from our experiments, a stochastic regular grammar has sufficient modeling power for discriminating TFBMs from cover-text.

2.2 Algorithm

The central problem of deciphering a stegoscript is learning a statistical model (an HMM shown in Fig. 1) from a given script. Since the problem is complex as a large number of motifs are to be discovered and many model parameters to be computed, we adopt an iterative learning technique to build a series of models, i.e., $\langle D_1, G_1 \rangle \Rightarrow \dots \Rightarrow \langle D_k, G_k \rangle \Rightarrow \dots$, where each model $\langle D_k, G_k \rangle$ contains motifs of lengths up to k (inclusive). In each iteration, $\langle D_k, G_k \rangle$ is optimized to best fit the script \mathcal{S} ; the optimized model is then used to identify over-represented motifs and words of length $k+1$, resulting in new model $\langle D_{k+1}, G_{k+1} \rangle$. The algorithm, which we call *WordSpy*, iterates through two major phases: *word sampling* and *model optimization*.

Starting with the simplest model $\langle D_1, G_1 \rangle$ with only one word of single base in D_1 , at the k -th iteration, *WordSpy* first identifies over-represented words of length k using its word sampling component. In this process, the given script \mathcal{S} is assumed to be generated by an optimized model $\langle D_{k-1}, G_{k-1} \rangle$, as it is the best model known so far. A word is over-represented if it occurs in \mathcal{S} more often than it could be generated by the current model $\langle D_{k-1}, G_{k-1} \rangle$. Some of the newly discovered words are merged to form degenerate motifs. All the new motifs are then merged with D_{k-1} to form the next dictionary D_k , and the model is retrofitted to accommodate the new motifs, leading to the next grammar, G_k . The new model $\langle D_k, G_k \rangle$ is then optimized in the model optimization component. The overall process repeats until the model covers motifs up to a maximum length. The algorithm can use additional information, such as gene expression profiling and TF location information such as that from CHIP-chip experiments, to specify the background words in the word sampling component.

Model Optimization. The model optimization component is for computing the next model $\langle D_k, G_k \rangle$ based on $\langle D_{k-1}, G_{k-1} \rangle$ to incorporate a set of new over-represented words of length k . The next dictionary D_k can be simply formed by combining D_{k-1}

and the new words; the main issue is then to compute the next grammar G_k . G_k has two types of parameters. The first are the transition probabilities, Ψ , corresponding to the word usage frequencies, determining which conserved motifs or background words are used in the given sequences. The second parameters are the emission probabilities, Θ , corresponding to the letter (base) usage frequencies at each position of an over-represented word. As these parameters are unknown, we apply an EM approach to estimate them.

We can write $G_k = (\Psi, \Theta, \mathcal{I})$, where $\Psi = \{P_B, P_M, P_{W_b}, P_{W_1}, P_{W_2}, \dots, P_{W_n}\}$ is the set of transition probabilities, $\Theta = \{\Theta_b, \Theta_1, \Theta_2, \dots, \Theta_n\}$ is a set of emission probabilities corresponding to the motifs and words in $D_k = \{W_b, W_1, W_2, \dots, W_n\}$, and $\mathcal{I} = \{I_{W_i} | W_i \in D_k\}$ is a set of indicators, where

$$I_{W_i} = \begin{cases} 1, & \text{if } W_i \text{ is a conserved motif,} \\ 0, & \text{if } W_i \text{ is a background word.} \end{cases}$$

I_{W_b} is always set to 0; the other values of \mathcal{I} are determined in the word clustering component. That is, \mathcal{I} does not change during model optimization. Without loss of generality, we view a set of sequences as a long sequence $\mathcal{S} = s_1 s_2 \dots s_q$. Let $\Psi^{(t)}$ and $\Theta^{(t)}$ be G_k 's parameters in the t -th iteration of the EM algorithm. The process of model optimization iteratively updates $\Psi^{(t)}$ and $\Theta^{(t)}$ until convergence.

To update $\Psi^{(t)}$ and $\Theta^{(t)}$, we first consider different parses of \mathcal{S} . The probability of a parse of \mathcal{S} , denoted by ϕ , given $\Psi^{(t)}$ and $\Theta^{(t)}$, can be computed by

$$P(\phi | \mathcal{S}, \Psi^{(t)}, \Theta^{(t)}) = \frac{\prod_{W_k \in D} \left(P_{W_k}^{(t)} \right)^{N_{W_k}^\phi} \prod_{i=1}^{N_{W_k}^\phi} P\left(\chi_{W_k}^i | \Theta^{(t)}\right)}{P(\mathcal{S} | \Psi^{(t)}, \Theta^{(t)})}$$

where $N_{W_k}^\phi$ is the count (i.e., number of occurrences) of motif W_k in the parse ϕ , and $\chi_{W_k}^i$ is the i -th occurrence (or site) of the W_k in \mathcal{S} under ϕ . Then the average number of occurrence of W_k , denoted N_{W_k} , can be calculated as

$$N_{W_k} = \sum_{\phi \in \Phi} P\left(\phi | \mathcal{S}, \Psi^{(t)}, \Theta^{(t)}\right) N_{W_k}^\phi \quad (1)$$

where Φ is the set of all possible parses. The average count of a letter ς at j -th position of W_k , denoted by $C_{W_k}(\varsigma, j)$, can be calculated by

$$C_{W_k}(\varsigma, j) = \sum_{\phi \in \Phi} P\left(\phi | \mathcal{S}, \Psi^{(t)}, \Theta^{(t)}\right) C_{W_k}^\phi(\varsigma, j) \quad (2)$$

where $C_{W_k}^\phi(\varsigma, j)$ is the count of letter ς at j -th position of W_k in the parse ϕ . Based on maximum likelihood principle, we update the parameters as follows.

$$\begin{cases} P_B^{(t+1)} = \frac{\sum_{W \in D} N_W \cdot (1 - I_W)}{\sum_{W \in D} N_W}, \\ P_M^{(t+1)} = \frac{\sum_{W \in D} N_W \cdot I_W}{\sum_{W \in D} N_W}, \\ P_{W_k}^{(t+1)} = \frac{N_{W_k}}{\sum_{W \in D} N_W \cdot \delta(I_{W_k}, I_W)}, \\ \Theta_k^{(t+1)}(\varsigma, j) = \frac{C_{W_k}(\varsigma, j)}{\sum_{\varsigma' \in \Sigma} C_{W_k}(\varsigma', j)}, \end{cases} \quad (3)$$

where Σ is the alphabet, $\varsigma \in \Sigma$, $j = 1, \dots, l(W)$, $l(W)$ is the length of W , and $\delta(x, y)$ equals 1 if $x = y$, or 0 otherwise.

The calculation of (1) and (2) could be costly if we enumerate all possible parses. We adopted the dynamic programming *forward-backward* algorithm [14] to compute the most probable state when observing $s_i \in \mathcal{S}$. More precisely, let π_i be the state of s_i , the probability of s_i being at the j -th position of a motif W under the current grammar can be computed as

$$P(\pi_i = W[j]|\mathcal{S}, G_k) = \frac{f(\mu) \cdot \rho_W \cdot \varrho_W(\mu + 1, \nu) \cdot b(\nu + 1)}{P(\mathcal{S}|G_k)},$$

where W is a degenerate word in D_k , $W[j]$ is the j -th position of W , $f(\mu)$ is the probability of observing \mathcal{S} up to s_μ (inclusive) given G_k , $\mu = i - k$, $\rho_W = P_W(I_W P_M + (1 - I_W)P_B)$, $\varrho_W(i, j) = P(\mathcal{S}_{[i:j]}|W)$, and $b(\nu + 1)$ is the probability of observing \mathcal{S} from $s_{\nu+1}$ (inclusive) down to the end of \mathcal{S} , and $\nu = i - k + l(W)$. Function $f(i)$ can be recursively computed as

$$f(i) = \sum_{W \in D} \rho_W \cdot \varrho_W(i - l(W) + 1, i) \cdot f(i - l(W)).$$

Similarly $b(i)$ can be computed as

$$b(i) = \sum_{W \in D} \rho_W \cdot \varrho_W(i, i + l(W) - 1) \cdot b(i + l(W)).$$

Evidently, $P(\mathcal{S}|G_k) = f(q) = b(1)$.

Suppose $P(\pi_i = W_k[j]|\mathcal{S}, \Psi^{(t)}, \Theta^{(t)})$ is the probability of observing s_i at the j -th position of a motif W_k given $\Psi^{(t)}$ and $\Theta^{(t)}$, equations (1) and (2) can be simply computed as

$$\begin{cases} N_{W_k} = \sum_{i=1}^q P(\pi_i = W_k[1]|\mathcal{S}, \Psi^{(t)}, \Theta^{(t)}), \\ C_{W_k}(\varsigma, j) = \sum_{i=1}^q P(\pi_i = W_k[j], s_i = \varsigma|\mathcal{S}, \Psi^{(t)}, \Theta^{(t)}), \end{cases}$$

where q is the length of \mathcal{S} .

The model optimization is done iteratively using equations in (3) until convergence. This procedure is the most time consuming part of the WordSpy algorithm. Nonetheless, the hash scheme of indexing a word w directly to the degenerate motifs that may emit w in the dictionary reduces the average computation of the forward-backward algorithm from $O(LN)$ to $O(L)$, with a penalty of space increment of $O(N)$, where L is the sequence length and N the size of the dictionary. The overall space complexity is $O(L + N)$.

Word Sampling. In the word sampling phase, WordSpy identifies all the over-represented words of length k based on the optimal model $\langle D_{k-1}, G_{k-1} \rangle$. To guarantee completeness, all possible words of length k in \mathcal{S} are tested. The algorithm scans the stegoscript \mathcal{S} once, tabulates, using a hashing scheme, all exact words of length k in \mathcal{S} , and computes their over-representativeness. A word is considered over-represented if it occurs more frequently in \mathcal{S} than it could be generated by the model $\langle D_{k-1}, G_{k-1} \rangle$.

We measure the over-representativeness by a Z -score. Let N_w be the number of occurrences of a word w in \mathcal{S} and random variable \hat{N}_w be the number of occurrences of w in a script with the same length as \mathcal{S} which were supposedly generated by model $\langle D_{k-1}, G_{k-1} \rangle$. Denote $E(\hat{N}_w)$ and $\sigma(\hat{N}_w)$ as the mean and standard deviation of \hat{N}_w . The Z -score of w is defined as $Z_w = (N_w - E(\hat{N}_w))/\sigma(\hat{N}_w)$. It is nontrivial to compute the statistics of random variable \hat{N}_w . Consider a word w of length k in a sequence of length L generated by model $\langle D_{k-1}, G_{k-1} \rangle$. There are various ways to produce w using the model, for example, by concatenating words of a single letter, or by merging a word's suffix with another word's prefix. To compute the expected number of occurrences of w , $E(\hat{N}_w)$, we define $\mathcal{A}_w(i)$ (and respectively $\mathcal{B}_w(j)$) to be the set of words in D_{k-1} whose suffixes (and respectively prefixes) match the first i (and respectively the last j) letters of w . The expectation $E(\hat{N}_w)$ can be computed as

$$E(\hat{N}_w) = (L - k + 1) \cdot \left(\sum_{i=1}^k A_w(i) \cdot \left(\sum_{j=i+1}^k P(w_{[i+1, j-1]} | \langle D_{k-1}, G_{k-1} \rangle) B_w(j) \right) \right),$$

where

$$\begin{cases} A_w(i) = \sum_{W_u \in \mathcal{A}_w(i)} P_{W_u} P(w_{[1, i]} | \Theta_u^{(i-)}), \\ B_w(j) = \sum_{W_u \in \mathcal{B}_w(j)} P_{W_u} P(w_{[j, k]} | \Theta_u^{(-j)}), \end{cases}$$

and $w_{[j, k]}$ represents the subsequence of w from its j -th to k -th positions, P_{W_u} is the transition probability of motif W_u , $\Theta_u^{(i-)}$ and $\Theta_u^{(-j)}$ are the emission probabilities of the last i and first j positions of Θ_u , respectively. The computation of $\sigma(\hat{N}_s)$ is complex and costly [15][16]. Following the practice in the existing methods, in our current implementation, we approximate $\sigma(\hat{N}_s)$ by $E(\hat{N}_s)$.

All the words with Z -scores greater than a threshold are considered over-represented. All over-represented motifs will be classified into background words or motif words with some evaluation methods. Three evaluation methods will be described below. The background word will be directly added to the background subdictionary. The motif words will be further clustered to form degenerate motifs. Let $C = \{w_1, w_2, \dots, w_m\}$ be a set of words of length k , sorted in a non-increasing order of their Z -scores. From the beginning to the end of list C , we take a word w_i as a seed and search the words in C that match with w_i by at least m letters, where m is determined so that the chance of two random words of length k having m matched letters is less than 0.001. All such matched words are then merged with w_i and subsequently removed from C . The procedure terminates after all seeds have been examined. This heuristic assumes that the degeneracy is uniform over all positions of a motif. However, TFBMs may have one or two core parts that are more conserved than their flanking sequences, which sometimes may be “do-not-care” positions. Fortunately, the current model $\langle D_{k-1}, G_{k-1} \rangle$ keeps all short, but over-represented motifs that may include those possible cores of longer motifs. We can also make a non-uniform seed by parsing a word in C through D_{k-1} , finding some cores (substrings), fixing the seed at those core positions, and allowing mismatches at the other positions. Both strategies have been implemented in the current algorithm.

At the end of word sampling phase, the new words and motifs are then merged with D_{k-1} to form the next dictionary D_k , and the model is retrofitted to accommodate

the new motifs, leading to the next grammar, G_k . The new model $\langle D_k, G_k \rangle$ is then optimized in the *model optimization* phase. The overall process repeats until the model covers motifs up to a maximum length.

2.3 Separating Real Motifs from Pseudo-motifs

WordSpy is designed to identify a complete list of putative motifs and usually gives hundreds of significant words. How to separate true motifs from background words of the covertext is also critical. As the covertext consists of random strings, a proper Z -score threshold can be used to filter out most background words. However, the regulatory regions of a genome are not really random. There exist many highly over-represented pseudo-motifs that make it harder to find real functional motifs. Fortunately, functional motifs have some properties that make them separable from spurious ones.

Specificity to the Target Promoters. A discovered motif cannot be considered as a genuine TFBM specific to the genes of interest if it is prevalent in other promoter regions of the genome. We utilize this property to discriminate real motifs from fake ones by a whole genome analysis. This is done by a Monte Carlo simulation of thousands of runs. In each run, a set of promoters are randomly selected from the genome and the occurrence of a motif is counted. Then a genome Z -score, shortened as Z_g -score, is calculated to measure the specificity of the motif to the target promoters from which it was discovered with respect to randomly selected promoters. A high positive Z_g -score is desired as it means the motif is unlikely to be a background words.

Gene Expression Coherence. Statistically a set of genes sharing a motif will have more similar expression profiles than a set of arbitrary genes. Therefore, we can measure the likelihood of a motif being biologically meaningful by the coherence of the expressions of all the genes whose promoters contain the motif. We use the average coherence of pairwise gene expression to measure the coherence of a set of expression profiles. We call this measure G -score, where G stands for genes. A higher G -score indicates a more biologically meaningful motif. The pairwise gene expression coherence can be measured in many ways, such as Euclidean distances and correlation coefficients. Here, we present our results using correlation coefficients. We also analyzed the expression coherence score in [17] and a normalized version of the G -score. Our results on yeast indicated that the simple correlation-coefficient G -score works slightly better than the other two.

3 Results

3.1 Decipher an English Stegoscript

We applied WordSpy to a stegoscript (about 268K letters) that had the first ten chapters (about 112K letters) of novel *Moby Dick* embedded within. This stegoscript was created by Bussemaker et al. [12]. Fig. 2(a) shows a tiny portion of the stegoscript, where the underlined text is the title and first two sentences of Chapter One. We ran WordSpy with different Z -score thresholds and to find words of maximum length of 15. We measured performance by the true positive rate (TPR), the percentage of true words discovered



Fig. 2. Deciphering novel *Moby Dick* from a stegoscript. (a) A small portion of the script; the underlined text is the title and first two sentences of Chapter One. (b) Deciphered *Moby Dick*. The identified background words are marked out by dots.

Table 1. Results on a stegoscript containing the first ten chapters of novel *Moby Dick* for Z-score threshold 6. Total 18930 words are in the original text. Total discovered words in the deciphered text are 16522. *Word match ratio* determines the least percentage of position matches for a true word to be considered correctly predicted. *True words discovered* gives the numbers of true words correctly predicted. *True positive rate* is the percentage of *true words discovered* over the total words in the original text. *False words reported* is the number of words falsely predicted based on different word match ratios. *False prediction rate* is the percentage of false words reported over the total words in the deciphered text.

Word match ratio (%):	10	20	30	40	50	60	70	80	90	100
True words discovered:	16047	16026	15899	15670	15529	15046	14584	14066	13500	13435
True positive rate (%):	84.7	84.6	83.9	82.7	82.0	79.4	77.0	74.3	71.3	70.9
False words reported:	238	259	387	617	761	1272	1787	2361	3003	3087
False prediction rate (%):	1.4	1.5	2.3	3.7	4.6	7.6	10.8	14.2	18.1	18.6

over all the words in the original text, and false prediction rate (FPR), the percentage of false predictions in the deciphered text. In measuring these rates, we considered different degrees of matches between a word in the original text and a predicted word in the deciphered text. For example, we consider it as a correct prediction if a recovered word has at least a certain percent of letters matched to the original word, which we call word match rate. As summarized in Table 1, TPR decreases and FPR increases as the word match rate increases. If we take the most stringent criterion of 100% word match rate, WordSpy is able to recover ~70% exact original words with a false prediction rate of ~19% using Z-score threshold 6. As a comparison, when the word match rate is 50%, TPR increases to ~82% while FPR decreases to ~4.6%.

A close examination showed that the FPR initially decreases and then stays relatively constant as the Z-score threshold increases (Fig 3(a)). When the Z-score threshold is high enough (>5.5), most falsely predicted words will be filtered out. On the other hand, the

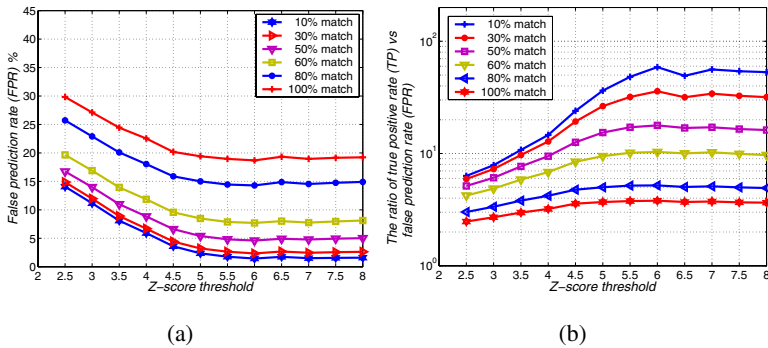


Fig. 3. Evaluation of WordSpy on a stegoscrypt of *Moby Dick*. (a) False prediction rates on different Z-score thresholds. (b) The ratios of true prediction rate over false prediction rate on different Z-score thresholds. The results are listed for different word matching ratios.

true positive rate (TPR) always decreases as the Z-score threshold increases. The overall best performance seems to be reached around the Z-score threshold of 6 (Fig. 3(b)).

To complete our example in Fig. 2(a), we show the recovered text in Fig. 2(b), where the identified background words are marked out with dots. This simple example interestingly shows that the deciphered text is pretty much readable.

3.2 Identifying Yeast Cell-Cycle TFBMs

To illustrate the power of WordSpy on real applications, we applied it to discover TFBMs of cell-cycle related genes of *S. cerevisiae* [18]. By removing spurious and homolog genes, we had 645 genes in the final set. Homologs were determined by WU-BLAST with E-value below 10^{-12} . The promoter sequences were retrieved using the RSA tools [24]. We compared WordSpy with the MobyDick algorithm [12], finding motifs with lengths upto 12. We tuned MobyDick to get its best possible parameters. The Z-score threshold for WordSpy was set to 3. The whole genome analysis on the specificity of the motifs, Z_g -scores, was performed with the promoters of all the genes of *S. cerevisiae*. We also used the yeast gene expression data [25] to calculate the G -score for each motif. As shown in table 2, all known cell-cycle related TFBMs were identified with high rankings in either Z_g -score or G -score. In contrast, MobyDick failed to discover three of them.

MBF and SBF are predominant TFs in the G1/S phase of the yeast cell-cycle. Their binding motifs, MCB (ACGCGT) and SCB (CRCGAAA) [26], are consistent with the top motifs discovered by WordSpy. Among 199 discovered motifs of length 7, AACGCGT is ranked first in both Z_g -score and G -score, CGCGAAA is ranked second in G -score and third in Z_g -score, and CACGAAA ranks 10th in Z_g -score and 17th in G -score. Another prominent motif GTAAACA (8th in Z_g -score and 10th in G -score) has been reported as the binding motif of Fkh2 (or Fkh1) [21], which is involved in cell-cycle control during pseudohyphal growth and in silencing of MHRa [27]. WordSpy also identified the binding motifs of Ace2/Swi5 and Met4/Met28 with high G -score rankings, and the binding motifs of Mcm1 and Ste12 with high Z_g -score rankings.

Table 2. Discovered known motifs in the promoters of 645 yeast cell-cycle genes. The first two columns list the known TFs and the known binding motifs. The next six columns report the results from WordSpy, followed by the last column of the results from MobyDick. The motifs discovered by WordSpy are marked with (+) if on the up strand, (-) if on the down strand or (*) if on both strands. The Rank is based on Z_g -score and G -score respectively, where the first number is the ranking and the second is the total number of discovered motifs of the same length.

Known TFs	Known motifs	WordSpy	Z-score	Z_g -score	Rank	G-score	Rank	MobyDick			
Ace2, Swi5	RRCCAGCR [1,8]	CCAGC (-)	5.4	5.2	8/29	0.0363	3/29	ACCCGCTGG			
		GCCAGC (+)	5.3	2.6	36/58	0.0551	4/58				
		AGCCAGC (+)	4.6	2.5	75/199	0.0688	13/199				
		CCAGCAAA (-)	4.3	3.5	107/867	0.113	51/867				
		CCAGCAAG (-)	3.9	2.9	185/867	0.0976	67/867				
		GCCAGCA (-)	3.9	3.4	124/867	0.1872	12/867				
		AGCCAGCA (+)	5.7	2.7	189/867	0.0929	73/867				
		AACCAGCA (+)	3.8	2.6	239/867	0.1983	8/867				
		Swi6, Mbp1	ACGCGT [1,8], [2,0]	AACGCGT (+)	13.7	11.3	1/199		0.1816	1/199	AACGCGT ACGCGTC
				GACGCGTC (+)	9.3	4.9	41/867		0.2106	4/867	
AAACGCGT (+)	14.6			10.2	3/867	0.2093	5/867				
AACGCGTC (*)	10.8			8.9	9/867	0.2003	7/867				
ACGCGTAA (*)	9.6			9.0	7/867	0.1341	36/867				
ACGCGTCA (*)	8.9			7.3	15/867	0.1291	41/867				
CAACGCGT (+)	6.3			4.0	73/867	0.1014	59/867				
Swi4, Swi6	CACGAAA [1,8], [2,0]			CACGAAA (*)	4.6	5.7	10/199	0.0623	17/199	CGCGAAA	
				ACACGAAA (-)	6.6	4.5	57/867	0.1081	55/867		
				CACGAAA (+)	7.1	5.5	32/867	0.1053	57/867		
	CGCGAAA [2,0]	CGCGAAA (*)	14.9	10.6	3/199	0.132	2/199				
		ACGCGAAA (*)	15.2	10.3	1/867	0.1733	15/867				
		CGCGAAA (+)	17.7	9.4	4/867	0.1352	34/867				
		Fkh1, Fkh2	GTAAACA [2,1]	GTAAACA (+)	8.2	7.4	8/199	0.084	10/199		GTAAACA
GGTAAACA (+)	7.2			4.6	48/867	0.1578	21/867				
GTAAACA (*)	9			6.6	11/867	0.098	66/867				
ATAAACAA (*)	8.8			5.9	23/867	0.0657	142/867				
MCM1	TTTCCTAA [2,1]	TTTCCTAA (+)	5.5	5.2	35/867	0.0435	307/867	N/A			
		Ste12	TGAAACA [2,2]	TTGAAACA (*)	4.3	4.2	66/867		0.0647	145/867	N/A
TGAAACA (*)	5			4.8	46/867	0.0631	149/867				
Met4, Met28 Chf1	TCACGTG [2,3]	TCACGTG (-)	5	1.7	129/199	0.0845	9/199	N/A			
		GTACACGTG (-)	5	0.9	661/867	0.2205	3/867				

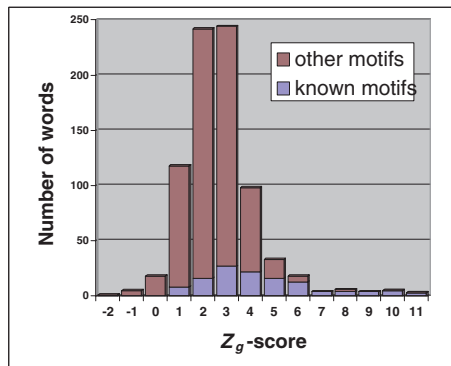


Fig. 4. Distribution of discovered yeast cell-cycle motifs of length 8 based on Z_g -score

Fig. 4 shows the distribution of all discovered motifs of length 8 based on Z_g -score. The motifs that match by at least 6 nucleotides with some known motifs are displayed

Table 3. The binding motifs found by three algorithms which are closest to the known TFBMs of the 12 yeast cell-cycle TFs Promoters were chosen based on Lee *et al.*'s ChIP-chip experiments. The rankings from each algorithm are included in parentheses. The rankings for WordSpy are among the words of the same length.

TFs	Known TFBMs	WordSpy		MEME	AlignACE	
ACE2	CCAGCA	GCTGG(1)	CCAGC(2)	GCTGGC(1)	AACCAGC(7)	AACCAGC(12)
Fkh1	GTAACA	GTAACA(1)	TGTTTAC(2)	GTAACAA(1)	TGTTTAC(2)	TAACAA(5)
Fkh2	GTAACA	GTAACA(1)	TGTTTAC(2)	GTAACAA(1)	TGTTTAC(2)	AANRWAAACA(3)
Mbp1	ACGGGT	ACGGGT(1)	AACGGT(1)	ACGGTT(2)	AACGGTT(2)	RACGGWY(3)
	CRCGAA	GACGGA(3)	TGGCTC(5)	ACGGAA(6)	n/a	ACGGWAAAA(9)
Mcm1	TTTCCTAATTAGGAAA	TAGGAAA(1)	TTTCCTAA(9)	TTAGGAAA(10)	CCTAATTAGG(1)	TTNCNNNTNNGGAAA(1)
Met4	TCACGTG	CACGTG(1)	TCACGTG(2)	ACTGTG(6)	CACGTG(1)	CACGTGAY(2)
	AAACTGTGG	GTGGC(1)	CCACA(3)	TGTGG(5)	CTGTG(6)	AAACTGTGG(4)
		TGTGGC(2)	CCACAGT(3)	GCCACAC(4)	ACTGTGG(5)	AAANTGTGGC(4)
Met31	AAACTGTGG	TGTGGC(1)	GCCACA(2)	GCCACAC(2)	ACTGTGG(7)	CACGTGANN(7)
	TCACGTG	CACGTG(1)	TCACGTG(3)	TGTGGCG(10)	AACTGTGG(7)	CACGTGANN(7)
Stb1	ACGGGA	AACGG(4)	TCGGTT(3)	TCGGTT(3)	TCGGTT(3)	AACGCSAAAA(3)
	CRCGAA	TTTCGG(1)	TTTCGG(1)	TTTCGG(2)	TTTCGG(5)	AACGCSAAAA(3)
	ACGGGT	ACGGT(3)			n/a	n/a
Ste12	TGAACA	GTAACA(1)	ATGAAAC(2)	TGAACAA(2)	TGAACA(2)	ATGMAAC(13)
Swi4	CGCGAAA	ACGGGAA(1)	GACGGA(2)	AAACGGC(3)	CACGAAA(7)	RACGCGAAA(2)
	ACGGGT	AACGGT(10)			n/a	n/a
Swi5	CCAGCA	GCTGG(1)	CCAGC(2)		n/a	n/a
Swi6	ACGGGT	ACGGT(1)	AACGGT(2)	ACGGTT(3)	AACGGTT(2)	AAACGGW(4)
	ACGGGA	AAACGGC(5)	CGCGTT(6)	ACGGAA(10)	TTTCGGC(12)	AAACGGW(4)

in a different color. This result demonstrates that most top-ranking motifs based on Z_g -score resemble known ones.

3.3 Finding Discriminative Motifs

Given two sets of scripts or sequences, a discriminative motif is such a motif that is over-represented in one script but not in the other. WordSpy is, in essence, an algorithm for finding discriminative motifs, because of its intrinsic feature of modeling motifs and background words in an integral model. Here, background words can be learned from one set of sequences, while the discriminative motifs are learned from another set given the background words.

We applied WordSpy as a discriminative algorithm to find TFBMs of *S. cerevisiae*. We constructed positive and negative sequence data based on the ChIP-chip experiments of Lee *et al.* [28]. For a specific TF, we selected as a positive dataset those promoters that the TF could bind to with p-values < 0.01 in the ChIP-chip experiments and as negative those promoters with p-values > 0.99 . We compared WordSpy with two widely used algorithms, MEME [3] and AlignACE [5]. MEME was executed with a 6-th order Markov model on the yeast non-coding regions as a background. Table 3 lists the motifs that are closest to the known cell-cycle related motifs from these three algorithms. As shown, WordSpy found all known motifs and some of their variations which may also be potential TFBMs. MEME was able to find most known motifs for each TF, but missed some binding sites of co-factors. AlignACE was also able to find many known motifs, while its predictions usually contained many false positives. It also missed the binding sites of some co-factors.

4 Discussion

We proposed a new approach to the challenging problem of genome-wide motif finding. Our approach combines a novel steganalysis method for discovering over-represented

motifs and methods for selecting biologically significant motifs. By approaching the motif-finding problem from a steganalysis perspective, we were able to accurately identify a large number of motifs of nearly optimal lengths. By finding motifs from the promoters of all the genes of interest, we avoided the problem of subjectively partitioning the genes into small clusters, which may make some motifs difficult to detect. By applying our approach to discover *cis*-acting elements from all cell-cycle related genes in yeast, we demonstrated its power as an effective genome-wide motif finding approach, which compared favorably to many existing methods.

The core motif-finding algorithm, WordSpy, combines both word counting and statistical modeling. Similar to word-counting methods, WordSpy can simultaneously detect a large number of putative motifs. However, different from the existing word-counting methods, the counting procedure of WordSpy is progressive and retrospective. It considers from short to long words and adjusts the over-representativeness of short words after having examined longer words, forcing not truly over-represented short words to be eliminated. As a result, WordSpy produces less spurious motifs and is able to find motifs with optimal lengths. Furthermore, instead of using statistical models to characterize a few unknown motifs with multiple local alignments, WordSpy models a large number of motifs, their compositions and usage to fit to the given sequences. Consequently, all significant words in regulatory regions can be identified.

WordSpy is a dictionary based approach, which was initiated in the innovative MobyDick algorithm by Bussemaker *et al.* [12]. Nevertheless, we have significantly extended their work in many important aspects. First, we took a novel steganographic viewpoint toward the problem of motif finding. This allows us to combine a grammar with a dictionary in a statistical model so as to accurately calculate word over-representativeness by computing its expected number of occurrences based on the best model built so far, while MobyDick computes the over-representativeness of a word by counting its occurrences in a large synthetic data. Second, WordSpy uses a hashing scheme and directly samples the words from the sequences without enumerating all possible words, which saves a substantial amount of computation. Third, we explicitly include a background word sub-model, which can take care of the pseudo-motifs, especially when negative sequence data are provided. Fourth, WordSpy builds and learns a stochastic dictionary of degenerate words with word clustering and HMM optimization, while MobyDick generates degenerate motifs by word enumeration. Another dictionary based approach by Gupta and Liu [29], called SDDA, also builds a stochastic dictionary. However, SDDA incrementally samples motifs one by one from randomly initialized matrices and is more like a Gibbs sampling based algorithm, suitable for detecting motifs from a small set of co-regulated genes.

In the current implementation of WordSpy, we assumed that the words in a dictionary were used independently. For some applications, however, the spatial correlations among motifs may be biologically important. For such cases, we may resort to a more complex grammar, such as stochastic context free or context sensitive grammar [13]. However, the incurred computational cost could be prohibitively high for even small problems. A more efficient way to capture motif correlations is to construct motif modules using motifs identified by a simple grammar model. Similar post-processing strategies [30] have been proposed.

In this research, we adopted two schemes to measure motif biological significance. One is the expression coherence of the genes whose promoter regions contain a motif, and the other is the specificity of a motif to the genes of interest with respect to the rest of the genome. As shown in this study, these two biological relevance measures are effective in identifying cell-cycle related TFBMs of buddy yeast.

In this study, we applied our approach to discover significant *cis*-elements from sequences of a single species. Our approach, especially the WordSpy algorithm, can be used to discover motifs conserved across multiple species, as in many comparative genomics approaches [31][32]. Nevertheless, computational tools for large-scale *de novo* motif finding for a single species are still important, especially for applications where no sequences of closely related species are available and problems where species-specific motifs are needed. It is interesting to note that single-species motif finding can be competitive comparing to comparative genomics methods using multiple species [33].

Acknowledgement

We are grateful to Gary Stormo and Hao Li for their insightful comments on this work. Thanks to Harmen Bussemaker, Hao Li, and Eric Siggia for their MobyDick program, and Jun Liu for their SDDA program. The research was funded in part by NSF grants EIA-0113618 and IIS-0535257.

References

1. B. Lemon and R. Tjian. Orchestrated response: A symphony of transcription factors for gene control. *Genes Dev.*, 14(20):2551–69, 2000.
2. C.E. Lawrence, S.F. Altschul, M.S. Bogouski, J.S. Liu, A.F. Neuwald, and J.C. Wooten. Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
3. T.L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using EM. *Machine Learning*, 21(1-2):51–80, 1995.
4. G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–77, 1999.
5. J.D. Hughes, P.W. Estep, S. Tavazoie, and G.M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Molecular Biology*, 296(5):1205–14, 2000.
6. J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Molecular Biology*, 281(5):827–42, 1998.
7. S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. In *8th Intern. Conf. on Intelligent Systems for Molecular Biology*, 2000.
8. M.Q. Zhang. Large scale gene expression data analysis: A new challenge to computational biologists. *Genome Research*, 9(8):681–8, 1999.
9. E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, 19:273–82, 2003.
10. Y. Tamada and et al. Estimating gene networks from gene expression data by combining bayesian network model with promoter element detection. *Bioinformatics*, 19:227–36, 2003.
11. P. Wayner. *Disappearing Cryptography*. Morgan Kaufmann, 2 edition, 2002.

12. H.J. Bussemaker, H. Li, and E.D. Siggia. Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *Proc. Natl. Acad. Sci. USA.*, 97(18):10096–100, 2002.
13. J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2 edition, 2001.
14. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
15. Mireille Regnier. unified approach to word statistics. In *RECOMB*, pages 207–213, 1998.
16. G. Reinert, S. Schbath, and M.S. Waterman. Probabilistic and statistical properties of words: An overview. *J. Computational Biology*, 7(1-2):1–46, 2000.
17. Y. Pilpel, P. Sudarsanam, and G.M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29(2):153–9, 2001.
18. P.T. Spellman and et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–97, 1998.
19. P. Dohrmann, W. Voth, and D. Stillman. Role of negative regulation in promoter specificity of the homologous transcriptional activators *ace2p* and *swi5p*. *Mol. Cell Biol.*, 16(4):1746–58, 1996.
20. J. Zhu and M.Q. Zhang. SCPD: A Promoter Database of Yeast *Saccharomyces cerevisiae*. *Bioinformatics*, 15:607–11, 1999.
21. M. Kato, N. Hata, N. Banerjee, B. Futcher, and M.Q. Zhang. Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology*, 5:R56, 2004.
22. J.W. Dolan, C. Kirkman, and S. Fields. The yeast STE12 protein binds to the DNA sequence mediating pheromone induction. *Proc. Natl. Acad. Sci. USA.*, 86(15):5703–7, 1989.
23. P.L. Blaiseau and D. Thomas. Multiple transcriptional activation complexes tether the yeast activator Met4 to DNA. *EMBO J*, 17:6327–36, 1998.
24. J. van Helden, B. Andre, and J. Collado-Vides. A web site for the computational analysis of yeast regulatory sequences. *Yeast*, 16(2):177–87, 2000.
25. J.M. Stuart, E. Segal, D. Koller, and S.K. Kim. A gene coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–55, 2003.
26. C. Koch, T. Moll, M. Neuberg, H. Ahorn, and K. Nasmyth. A role for the transcription factors Mbp1 and Swi4 in progression from G1 to S phase. *Science*, 261:1551–7, 1993.
27. P.C. Hollenhorst, M.E. Bose, M.R. Mielke, U. Müller, and C.A. Fox. Forkhead genes in transcriptional silencing, cell morphology and the cell cycle: Overlapping and distinct functions for FKH1 and FKH2 in *Saccharomyces cerevisiae*. *Genetics*, 154:1533–48, 2000.
28. T.I. Lee and et al. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298:799–804, 2002.
29. M. Gupta and J. Liu. Discovery of conserved sequence patterns using a stochastic dictionary model. *J. Amer. Statist. Assoc.*, 98:55–66, 2003.
30. S. Sinha, E. V. Nimwegen, and E. D. Siggia. A probabilistic method to detect regulatory modules. *Bioinformatics*, 19:292–301, 2003.
31. M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E.S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–54, 2003.
32. W.W. Wasserman, M. Palumbo, W. Thompson, J.W. Fickett, and C.E. Lawrence. Human-mouse genome comparisons to locate regulatory sites. *Nature Genetics*, 26(2):225–8, 2000.
33. E.D. Siggia. Computational methods for transcriptional regulation. *Cur. Opin. Gene. and Deve.*, 15:214–21, 2005.

Causal Inference of Regulator-Target Pairs by Gene Mapping of Expression Phenotypes

David Kulp and Manjunatha Jagalur

Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
{dkulp, manju}@cs.umass.edu

Abstract. Correlations between polymorphic markers and observed phenotypes provide the basis for mapping traits in quantitative genetics. When the phenotype is gene expression, then loci involved in regulatory control can theoretically be implicated. Recent efforts to construct gene regulatory networks from genotype and gene expression data have shown that biologically relevant networks can be achieved from an integrative approach. Inspired by epistatic models of multi-locus QTL mapping, we propose a unified model of expression and genotype representing *cis*- and *trans*-acting regulation. We demonstrate the power of the model in contrast to standard interval mapping by automatically discovering specific pairs of regulator-target genes in yeast. Our approach's generality provides a convenient framework for inducing a regulatory network topology of directed and undirected weighted edges.

1 Introduction

Conventionally, quantitative geneticists aim to identify one or usually at most two loci associated with a single phenotype. By contrast, computational biologists have leveraged large numbers of gene measurements — most notably whole genome gene expression — to infer entire networks of associated (correlated) genes (Friedman et al. (2000); Pe'er et al. (2001); and many others). Recently several data sets of both whole genome genotype and expression data have been published (Morley et al., 2004; Schadt et al., 2003; Steinmetz et al., 2002; Brem et al., 2002). In a major departure from conventional genetic trait analysis, now 10^5 to 10^6 phenotypic traits are represented by distinct gene expression measurements. Theoretically, chromosomal regions can be linked to the expression of each of the many measured genes. Thus, this data provides the basis for determining the role of genetic variation in differential gene expression and the identification of polymorphic genes that regulate, directly or indirectly, transcriptional control.

1.1 Background: Genetics of Gene Expression

The three most significant studies to date include whole genome expression and genotyping for a collection of 112 segregants from a mating of two isogenic yeast

strains (Brem and Kruglyak, 2005), a collection of 111 selfed progeny from a mating of two inbred mice strains (Schadt et al., 2003), and 32 recombinant inbred mice strains (Chesler et al., 2005). Since the yeast regulatory system is simpler and recombination rates are higher than mouse, we focus on the work of Brem and colleagues exclusively. In that work, expression levels were measured for 5727 ORFs and genotype data obtained for 2957 markers regularly spaced across the yeast chromosomes.

Quantitative trait loci (QTL) are regions on a genome in which the genetic variation is significantly correlated with a phenotypic trait, here the expression of a gene. Conventional mapping models assume a linear relationship of genotype to phenotype according to a mixture model of the form:

$$P(T_i|Q_j) = \mathcal{N}(\beta_0 + \beta_1 Q_j, \sigma)$$

where T_i is the expression of the i^{th} transcript and Q_j is a numeric genotype value corresponding to the j^{th} site in the genome. (Genotypes are usually assumed to be biallelic with values of -1 , 0 , and 1 . Backcrosses and genotypes of haploid phase are simpler with values of 0 and 1 .) For a fixed T_i , all possible Q_j are considered and the β terms fit by regression. When measured markers are sparse, Q_j is usually sampled in regular intervals between the markers, in which case, Q_j is estimated according to a maximum likelihood model called interval mapping (Lander and Botstein, 1989). Consecutive regions along a chromosome with log odds (LOD score = $\log_{10} \frac{P(T_i|Q_j, \beta_0, \beta_1, \sigma)}{P(T_i|Q_j, \beta_0, \beta_1=0, \sigma)}$) greater than some threshold are identified as candidate QTL intervals, within which are genes believed responsible for the phenotypic variation. In our case, each QTL interval putatively contains a regulating gene.

Brem et al. (2002) found that for yeast a large fraction of differential gene expression was due to genetic variation and Yvert et al. (2003) showed that, perhaps surprisingly, the genes in the QTL intervals were not enriched for transcription factors or any particular gene function. This observation could possibly be explained by the large size of the QTL intervals, typically containing ten or more genes, however we will show that our more precise models still do not implicate transcription factors as dominant upstream regulators. Thus, one must conclude that regulatory influence is a complex process such that “upstream regulator” is interpreted in the broadest of contexts.

Polymorphisms affecting gene expression are conventionally divided into *cis*- and *trans*-acting effects, i.e. polymorphisms that are proximal to the gene, such as in the promoter or 3' end, and those in another gene. Detecting *cis*-acting QTLs is straightforward using interval mapping and Schadt et al. (2003) showed that such QTLs tend to be highly significant. On the other hand, Brem and Kruglyak (2005) note that *trans*-acting variation accounts for most differential gene expression, but that a large number of weak actors is common. This is consistent with a model of gene regulation in which multiple factors contribute in macromolecular complexes and in many different stages of cell regulation such as signaling, transport, and so on.

1.2 Background: Inferring Regulatory Networks from Correlated Gene Expression

Independent of the data sets described so far, large collections of gene expression over time course (Spellman et al., 1998) or varying environmental conditions (Gasch et al., 2000; Hughes et al., 2000) have been studied to reveal dependent variation among genes and thereby deduce regulatory relationships. A dominant model used in such analyses was first proposed by Friedman et al. (2000) in which each gene is a random variable with conditional distribution dependent on a small number of parent variables according to the Bayesian network (BN) formalism. Such models are based on the assumption that the data represents perturbations of an integrated system. (The genotype/expression data sets may also be seen as perturbations in which the DNA is perturbed by sampling matings.)

In the BN modeling method, the key design factors are (1) the estimation of the conditional probability term $P(T_i|Pa(T_i))$ — abstractly a score function, where $Pa(T_i)$ are the parents of gene T_i — and (2) an efficient means of discovering the set $Pa(T_i)$. Both parametric continuous and non-parametric discrete score functions have been considered. The discrete case is common in the literature; relative gene expression is discretized usually into categorical increased, decreased, and unchanged values and conditional probability tables (CPT) are constructed from tallied observations of the values among parents and child. A CPT model can theoretically capture complex relationships among the parents, but this power is usually limited by the binning of expression values into a few values in order to achieve adequate conditional density estimates.

Continuous models are attractive because parameters are estimated from the totality of the data, but computational efficiency concerns have conventionally limited the class of models considered to simple linear Gaussian models with a small number of parameters of the form

$$P(T_i|Pa(T_i)) = \mathcal{N}(\beta_0 + \sum_{j \in Pa(T_i)} \beta_j T_j, \sigma) .$$

Each parent adds an independent contribution to T_i , eliminating the potential interacting effects among parents. But interacting effects can be important under limited circumstances and we will suggest below a more general Gaussian model.

The major drawback of the BN approach for analyzing gene expression data alone is that the dependencies inferred among variables does not imply causality. Indeed, for any BN solution there is an equivalence class of alternative solutions with different edge directions that have the same joint probability.

1.3 Background: Previous Work

Zhu et al. (2004) describes a method to build comprehensive BN reconstructions of regulatory networks based on genotype/expression data sets. Their work is motivated, as we are, by the recognition that correlation between genotype

and expression *does* imply causality. Genotype assignments represent random shuffling during meiosis, so correlations observed must be the effect of causative polymorphisms. The approach by Zhu et al. (2004) is to weight the BNs of Friedman et al. with priors according to rules regarding the chromosomal positions of genes and the differences in QTLs between pairs of genes.

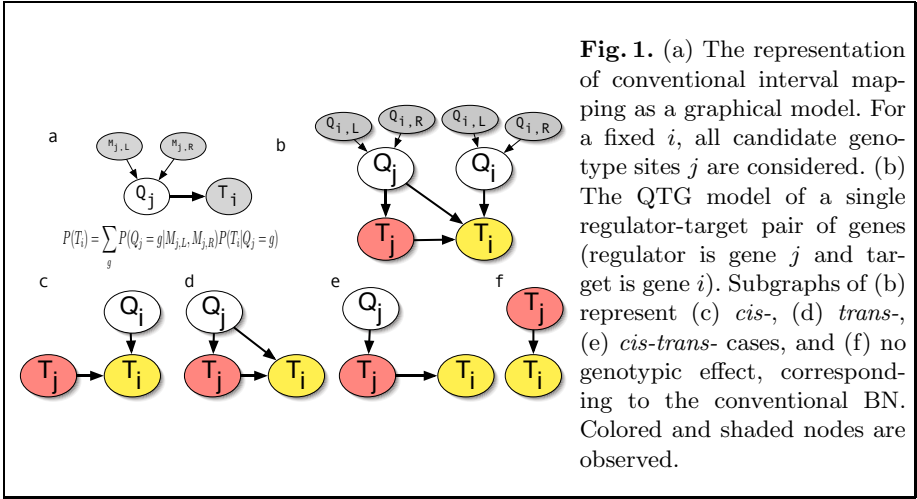
Li et al. (2005) also addressed this problem by filtering the set of candidate parent genes of a target gene to only those genes with coding SNPs located within QTL intervals with stringent LOD scores. With a much smaller set of possible model configurations it was possible to exhaustively search all BN configurations using a score function based on gene expression alone. In a similar strategy, Bing and Hoeschele (2005) recommend an analysis protocol for genotype/expression data in which individual genes within a QTL interval are considered as parents according to their expression correlation to the target gene.

In all of the above cited works, regulatory relationships are derived according to a two-step process in which standard QTL interval mapping is first applied — which serves to filter or prejudice the set of parent genes — followed by a selection of regulator-target pairs according to gene expression correlations. It is conceivable that the interaction between polymorphisms in and expression of a regulator may have a significant effect, not observed by either factor alone. This argument for the inclusion of interacting effects is closely related to epistatic models for multiple QTL (Sen and Churchill, 2001). In the next section, we propose such a simple, unified model for the scoring of candidate regulator-target pairs that considers all scenarios of *cis*- and *trans*- effects, allowing for interaction among gene expression and genotype.

2 Methods

We represent the genotypes and the expression measures as numeric random variables in a graphical model. In the general case of QTL interval mapping using sparse marker data, the genotype at a site of interest is an unknown random variable, Q_j , dependent on the observed genotypes of the nearest upstream and downstream flanking markers, $M_{j,L}, M_{j,R}$. The conditional probability of the unobserved genotype is a well-known function of the recombination distances among Q_j , and $M_{j,L}, M_{j,R}$ (Lynch and Walsh, 1998). Assuming that some observed phenotype (here gene expression, T_i , where i ranges over the number of genes) is dependent on Q_j , then the graphical model is shown in figure 1a. QTL interval mapping is then the maximum likelihood estimate of each Q_j and the selection of those Q_j where the log likelihood exceeds some threshold.

We are concerned with the class of *trans*-acting regulators in which the expression of the target is dependent on the expression of the regulating gene. When incorporating variation, we consider three sub-classes of genotypic effect: *cis*-, *trans*-, and *cis-trans*-acting sites. For example, a variation in the promoter region or 3' end of the target gene may have a *cis*-acting effect on the expression level of the target; a variation in the coding region of the regulator may have a *trans*-acting effect, either directly or indirectly, on the expression of a target



gene, such as through the modification of a DNA-binding motif in a transcription factor; and variation in or around the regulator gene may have a *cis*-acting effect on the regulator’s expression which indirectly affords a *trans*-acting effect on the target, i.e. *cis-trans*.

If we consider only the genotype sites at the locations of the protein-coding genes in a fully annotated genome, then we can conveniently reference both genotypes and genes with a common index, i.e. Q_i represents the genotype for the gene i with expression T_i . Figure 1b naturally follows. We refer to this model as the full **QTG** model for a single *quantitative trait gene* and the process of estimating regulatory genes for a given target as “QTG mapping”. The three genotype sub-classes are subgraphs of the full model shown in figure 1c-f.

2.1 Case: *trans*-acting Regulator

In this paper we address only the *trans*-acting regulator sub-class of figure 1d where the target is dependent on both the genotype and expression of the regulator. It is important to recognize that this is a biologically reasonable scenario with many relevant examples in the data. For example, the scatter plots in figure 2 show the relationships among the expression of a target gene and the expression and genotype of putative regulators. In these cases only the combination, and sometimes interaction, of the regulator’s genotype and expression can adequately model the target expression.

Therefore, to consider the possible interactions among genotype and expression, our full model is

$$P(T_i | Q_j, T_j, \theta) = \mathcal{N}(\beta_0 + \beta_1 T_j + \beta_2 Q_j + \beta_3 T_j Q_j, \sigma) \quad (1)$$

where θ is the β and σ model parameters.

As with standard interval mapping, Maximum likelihood estimation can be achieved using an expectation maximization (EM) approach in which the

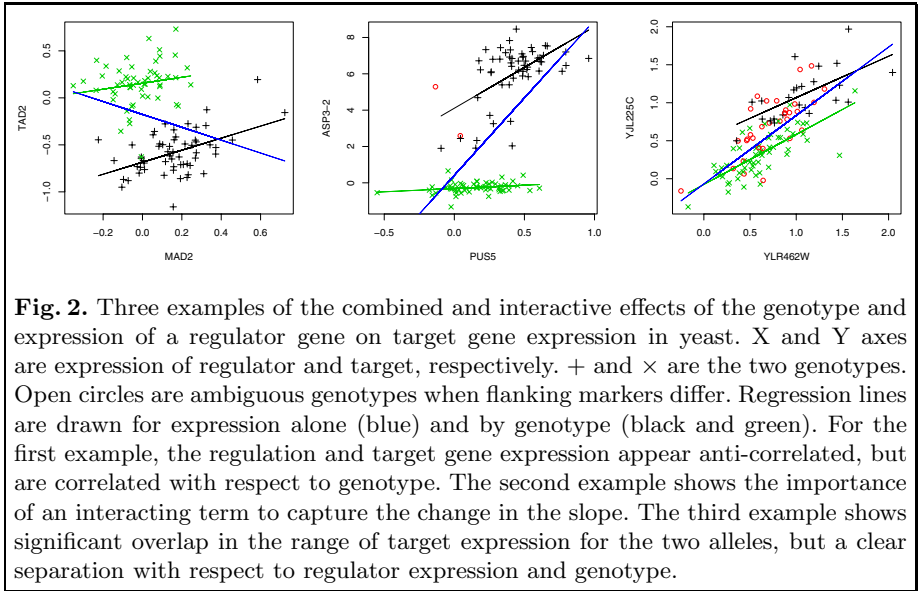


Fig. 2. Three examples of the combined and interactive effects of the genotype and expression of a regulator gene on target gene expression in yeast. X and Y axes are expression of regulator and target, respectively. + and × are the two genotypes. Open circles are ambiguous genotypes when flanking markers differ. Regression lines are drawn for expression alone (blue) and by genotype (black and green). For the first example, the regulation and target gene expression appear anti-correlated, but are correlated with respect to genotype. The second example shows the importance of an interacting term to capture the change in the slope. The third example shows significant overlap in the range of target expression for the two alleles, but a clear separation with respect to regulator expression and genotype.

genotype, Q_j , and the variables, θ , are alternatively estimated until convergence. But the advantage of this model over the standard mapping and multi-step approaches previously proposed is that individual loci are automatically mapped in a single step by simultaneously considering all available evidence.

Note that the strength of the genotypic effect is directly related to our ability to infer causality. That is, as the contribution of the β_2 and β_3 terms decreases, our confidence in the causal direction between genes i and j is reduced. We can be precise about this directionality by comparing our model with the simpler model of no genotypic effect (figure 1f). From equation (1), for each tested gene pair, i and j , we can determine the strength of a relationship (the *full model score*) as

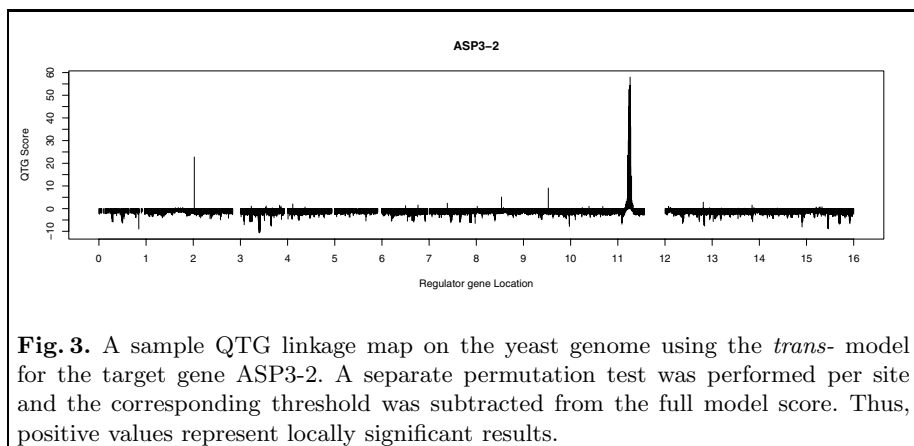
$$\log_{10} \frac{P(T_i|Q_j, T_j, \theta)}{P(T_i|Q_j, T_j, \theta : \beta_1 = \beta_2 = \beta_3 = 0)} \quad (2)$$

and the directionality (*genotype reduced-model score*) according to

$$\log_{10} \frac{P(T_i|Q_j, T_j, \theta)}{P(T_i|Q_j, T_j, \theta : \beta_2 = \beta_3 = 0)} \quad (3)$$

Moreover, if the β_2 and β_3 terms are weak, then it indicates that the major effect is the QTL interval and so our confidence in the *specific* regulator gene is correspondingly weak. Thus, confidence in the gene, T_j , as the actor in the relationship is found with the *expression reduced-model score*

$$\log_{10} \frac{P(T_i|Q_j, T_j, \theta)}{P(T_i|Q_j, T_j, \theta : \beta_1 = \beta_3 = 0)} \quad (4)$$



And so a hypothesis of a causal regulator-target relationship requires significant values from the full model and reduced-model scores (equations 2, 3, and 4). However, interesting results can be achieved when some, but not all, scores are significant. For example, we can identify dependent, but not causal, relationships when the genetic component is weak.

This leads us to propose a seed-based partially directed graph of regulatory relationships built from confident Markov pairs, much like the regulatory modules studied by Pe'er et al. (2001) and others. Deriving such a sub-network is computationally tractable and biologically relevant, since most biological analysis is concerned with a pathway centered around a gene of interest. Perhaps more importantly, inferring a complete network is not reasonable for any single data set where the perturbations are incomplete and the data is too sparse. In particular, with the genotype/expression data sets, regardless of the number of matings, differential expression can only be detected for a gene in which polymorphisms exists between parental strains in its regulatory pathway.

Our model can be used to produce a QTG map (figure 3) for each target gene, similar to conventional QTL maps, and a network of dependent genes as nodes whose edges are directed when causal influence is significant and undirected otherwise (figure 5).

3 Experimental Results

As with networks derived from gene expression alone, the connectivity does not necessarily imply physical interactions between genes. Yvert et al. (2003) previously observed that genes within QTLs computed from standard interval mapping were not enriched for any function. Nevertheless, we wondered whether this lack of functional enrichment was due to the imprecise mapping of intervals that contain usually tens of candidate genes. We hypothesized that our QTG mapping method, which identifies specific candidate genes, might show enrichment for transcription factors or other functional category.

To test this hypothesis, we analyzed the Brem and Kruglyak (2005) yeast set consisting of 6215 gene expression measurements and 2957 genotype markers across 112 matings between two distinct isogenic strains. We computed the pairwise dependency among genes according to the genotype reduced model score (equation 3) and selected only those relationships with a significance better than $p < 0.01$ based on a permutation test in which the target gene expression values were shuffled 1000 times. For simplicity in our analysis we assumed that each gene regulated or was regulated by at most ten genes. Among this set of relationships, we wished to consider those pairs of genes with confidence in both the directionality and the specific regulator gene. Thus, we selected those relationships where the full model and both reduced models were also significant ($p < 0.01$) based on permutation testing.

We then considered the significance of each Gene Ontology (GO, Ashburner et al. (2000)) category with respect to the known GO assignments to the candidate regulators using the standard hypergeometric distribution test. We found no significant enrichment for any molecular function or biological process.

Even though no functional enrichment in transcription factors was found, we still examined the predicted targets of transcription factors for evidence of physical interaction. Considering all the predicted targets of each transcription factor that met the selection criteria above, we searched 500nts upstream of the target for matches to known binding site motifs (TRANSFAC, Matys et al. (2003)). We found no significant enrichment for targets containing known binding regardless of sequence similarity thresholds. For example, only 35 of 719 putative targets contained matches to known binding sites. And of those, only 8 were known targets of their respective transcription factor regulators. This observation further confirms that regulatory behavior captured in genotype/expression networks is not likely to be physical interactions, but more complex, indirect relationships.

Next we wondered how well a causal relationship could be inferred when the regulator was part of a multifactor regulon. Using the yeast data set of $n = 6000$ genes, we simulated an $n + 1$ target gene according to an additive model of $k = 2 \dots 5$ regulators, with only one regulator having genotypic effect. Specifically, we simulated

$$T_{n+1} = \beta_1 T_1 + \dots + \beta_k T_k + \beta_{k'} T_k Q_k + \epsilon$$

where $\beta_{k'}$ was set at random values such that the genotypic effect between the two alleles, $(\mu_a - \mu_b)/\sigma$, was uniformly selected between 0.5 and 3.0. The other β 's were selected from $\mathcal{N}(0, 1)$. Using the QTG *trans* model we attempted to recover the *causal* regulator of the simulated target among the background of the other n genes. By modifying the full model threshold for equation 2 we obtained different trade-offs between recall and precision. We compared this approach to the standard QTL mapping approach and to a more liberal, but realistic, test in which we defined a true positive as correctly indentifying the *interval* containing the regulator (false positives were intervals not containing the true regulator). We found that our QTG model was successful in identifying the correct regulating gene, even for larger values of n (figure 4). Not surprisingly, conventional

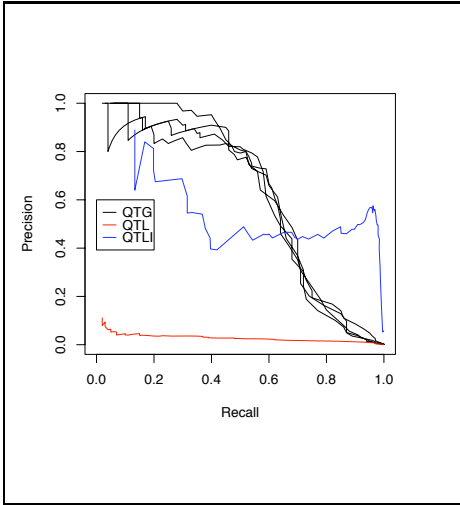


Fig. 4. A plot of recall ($\frac{TP}{TP+FN}$) versus precision ($\frac{TP}{TP+FP}$) for varying full model scores. **QTG**: our model; a positive classification is a regulator whose score exceeds the threshold; Multiple plots for QTG are shown for $k = 2 \dots 5$ as the number of additional regulators in the regulon, i.e. extra noise terms. **QTL**: the conventional QTL score where a positive is measured as with QTG; **QTLI**: an easier test using the QTL score where a true positive is called when the true regulator gene is found within the QTL interval.

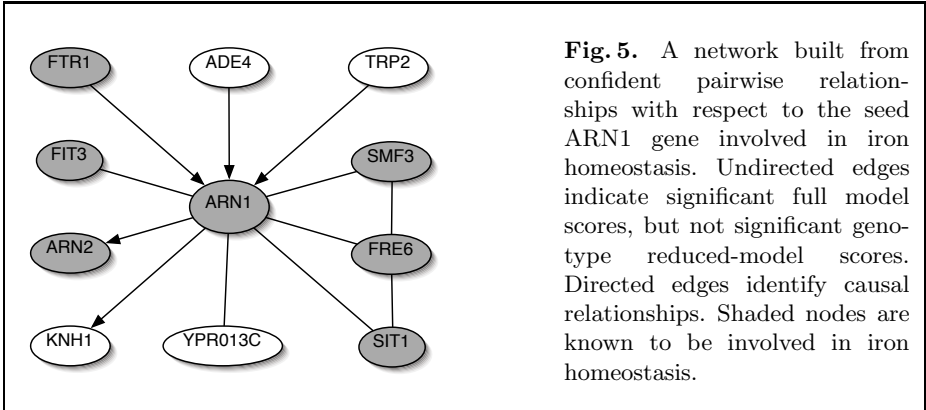
QTL mapping alone, being a function of only the flanking markers, failed to accurately predict the precise regulating gene, but the QTL interval was typically identified with reasonable success in our simulation.

To assess how well our model could identify true regulator genes, we considered six candidate QTL intervals analyzed in Brem et al. (2002). The intervals were each predicted as containing a regulator gene associated with a large number of target genes, although the precise gene was unknown. In the paper, a putative gene within each interval was predicted manually by the authors according to published gene annotations of regulator and targets. Using only the genotype/expression data, we automatically predicted a single gene within each interval by selecting the gene with the highest average full model QTG score over all the target genes linked by the authors to the interval (Appendix A). In five of the six cases our top or second-best prediction agreed with the manual prediction. In the one differing case two significant alternative genes were predicted by QTG, but neither appears to be functionally related to the set of target genes.

These six loci were originally identified by Brem et al. from 40 segregants, but since then an additional 72 individuals have been assayed by the authors. Interestingly, with the additional data, most of these QTL are no longer significant by genetic linkage alone, but our analysis still reveals highly significant individual regulator genes within these intervals.

Finally, we constructed a putative network from the seed gene, ARN1, using the Brem et al. data as a demonstration of the characteristics of a regulatory module that might be derived from our models. Regulatory modules of the iron homeostasis pathway have been previously constructed from the gene expression data of Hughes et al. (2000) by Pe'er et al. (2001), Pena et al. (2005), and Margolin et al. (2004).

For the purposes of this work, we consider networks constructed simply according to the pairwise relationships; that is, the network is not a BN, but its



edges are derived from Markov pairs according to our derived score functions. In future work we will discuss efficiently constructing a Markov blanket of the full BN for a seed gene, which is more complex than the basic approach for generic BNs.

Most genes in our module reconstruction in figure 5 are iron homeostasis and many that were found were common to previous reconstructions based on a different expression data set. Both the reductive mechanism directly associated with ARN1 (ARN2, SIT1, FTR1, SMF3, FIT3) and the non-reductive transport (FRE6) mechanism were implicated in the network. Interestingly, two key genes in this pathway, FTR1 and FET3, although not directly linked to ARN1, are both involved in iron uptake and were found to have significant *bi*-directional causality, implying an auto-regulatory mechanism.

4 Conclusion

We proposed an improved, principled method for mapping causal loci involved in transcriptional control when analyzing data sets of whole genome genotype and expression data. Our “QTG” model is a natural application of epistatic models to gene expression, allowing for interactions among gene expression and genotype. From a genetics perspective, our model is a more complex scoring function for the identification of a single, causative gene instead of the conventional multi-gene locus. And from a bioinformatics perspective, it is an improved score function for BN regulatory module reconstruction. We considered the simplest question of detecting relationships between regulator-target pairs, but we plan to extend this to the construction of Markov blankets.

Moreover, here we only presented results for the sub-model describing *trans*-acting genes, but the generality of the QTG model allows for incorporation of *cis*-acting effects for targets and regulators. Of note, when all polymorphic sites are known, but the haplotype is unknown for any sample, as is the case when crossing two fully sequenced genomes, then one may choose the full model or a sub-class of the full model as appropriate. For example, if there is no polymorphic

sites in the non-coding regions of the target gene then the *cis*-acting model parameters can be dropped since the genotype is uninformative.

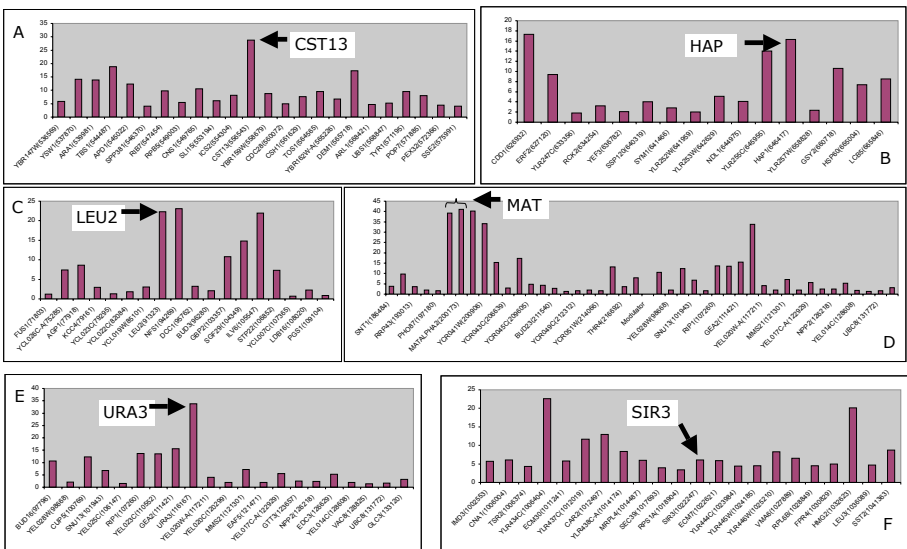
Despite the greater precision of the QTG model, we did not find any functional enrichment of regulators, confirming previous work using the simpler QTL model. But the value of the QTG approach is apparent in our initial results, which suggest that biologically meaningful pathways can be reconstructed and true regulators recovered from genotype/expression data sets.

Bibliography

- M. Ashburner, C.A. Ball, et al. Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–9, 2000.
- N. Bing and I. Hoeschele. Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, 170(2):533–42, 2005.
- R.B. Brem and L. Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proc Natl Acad Sci U S A*, 102(5):1572–7, 2005.
- R.B. Brem, G. Yvert, et al. Genetic dissection of transcriptional regulation in budding yeast. *Science*, 296(5568):752–5, 2002.
- E.J. Chesler, L. Lu, et al. Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nat Genet*, 37(3):233–42, 2005.
- N. Friedman, M. Linial, et al. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–20, 2000.
- A.P. Gasch, P.T. Spellman, et al. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–57, 2000.
- T.R. Hughes, M.J. Marton, et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–26, 2000.
- E.S. Lander and D. Botstein. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*, 121(1):185–99, 1989.
- H. Li, L. Lu, et al. Inferring gene transcriptional modulatory relations: a genetical genomics approach. *Hum Mol Genet*, 14(9):1119–25, 2005.
- M. Lynch and B. Walsh. *Genetics and analysis of quantitative traits*. Sinauer, Sunderland, Mass., 1998. 97017666 Michael Lynch, Bruce Walsh. Includes bibliographical references (p. 891-[948]) and indexes.
- A. Margolin, N. Banerjee, et al. http://www.menem.com/~ilya/digital_library/mypapers/margolin-et-al-b-04.pdf, 2004.
- V. Matys, E. Fricke, et al. TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*, 31(1):374–8, 2003.
- M. Morley, C.M. Molony, et al. Genetic analysis of genome-wide variation in human gene expression. *Nature*, 430(7001):743–7, 2004.
- D. Pe’er, A. Regev, et al. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:S215–24, 2001.
- J.M. Pena, J. Bjorkegren, et al. Growing Bayesian network models of gene networks from seed genes. *Bioinformatics*, 21 Suppl 2:ii224–ii229, 2005.
- E.E. Schadt, S.A. Monks, et al. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422(6929):297–302, 2003.
- S. Sen and G.A. Churchill. A statistical framework for quantitative trait mapping. *Genetics*, 159(1):371–87, 2001.

- P.T. Spellman, G. Sherlock, et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9(12):3273–97, 1998.
- L.M. Steinmetz, H. Sinha, et al. Dissecting the architecture of a quantitative trait locus in yeast. *Nature*, 416(6878):326–30, 2002.
- G. Yvert, R.B. Brem, et al. Trans-acting regulatory variation in *Saccharomyces cerevisiae* and the role of transcription factors. *Nat Genet*, 35(1):57–64, 2003.
- J. Zhu, P.Y. Lum, et al. An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenet Genome Res*, 105(2-4):363–74, 2004.

A Additional Figures



Six QTL intervals from the yeast genome in which multiple targets (from 7 to 28) were linked to a common interval. Genes within the intervals are shown by their order along the chromosome, but uniformly spaced across the X axis for visualization. The height of the bar for each gene represents the average QTL score among those genes linked to the loci. An arrow labels the putative regulator annotated by Brem et al. (2002). The annotated regulator in the interval is either the best or second best QTL score for all regions except *F*.

Examination of the tRNA Adaptation Index as a Predictor of Protein Expression Levels

Orna Man^{1,2}, Joel L. Sussman¹, and Yitzhak Pilpel²

¹Department of Structural Biology, The Weizmann Institute of Science, Rehovot 76100, Israel
{orna.man, joel.sussman}@weizmann.ac.il
<http://www.weizmann.ac.il/~joel>

²Department of Molecular Genetics, The Weizmann Institute of Science, Rehovot 76100, Israel
pilpel@weizmann.ac.il
<http://longitude.weizmann.ac.il>

Abstract. Phenotypic differences between closely-related species may arise from differential expression regimes, rather than different gene complements. Knowledge of cellular protein levels across a species sample would thus be useful for the inference of the genes underlying such phenotypic differences. dos Reis et al [1] recently proposed the tRNA Adaptation Index to score the optimality of a coding sequence with respect to a species' cellular tRNA pools. As a preliminary step towards a multi-species analysis that would utilize this index, we examine in this paper its performance in predicting protein expression levels in the yeast *S. cerevisiae* and find that it likely predicts maximal potential levels of proteins. We also show that tAI profiles of genes across species carry functional information regarding the interactions between proteins.

1 Introduction

A major challenge in evolutionary research is to understand molecular and genomic causes of phenotypic divergence of species. One obvious source of difference in phenotype and life-style among related species may be differences in their gene complements. The phylogenetic profiles method [2] utilizes this concept by clustering together genes that share the same pattern of presence/absence in the genomes of a set of species. A striking example [3] is the recent identification of an entire set of genes involved in the formation of cilia - short hair-like appendages found on the surfaces of some types of cells in some organisms. The genes were identified on the basis of their presence in all (sequenced) species known to have ciliated cells and absence in all (sequenced) species known to be devoid of this sub-cellular structure.

Such a methodology, although useful, is necessarily limited to genes that are found in only a fraction of the species analyzed, and may be problematic when one would like to make inferences for closely related species. Martin et al [4] created a matrix denoting for each *E. coli* open reading frame (ORF) its conservation, in a variety of prokaryotic species, relative to the *E. coli* sequence. They suggested that clustering this matrix by genes could lead to genotype-to-phenotype associations, and could perhaps even reveal the genes responsible for specific traits. As an example they identified functions that are over-represented in genes differentiating between Gram-positive and Gram-negative bacteria.

Recently, it has been shown that predicted levels of expression of functionally related proteins tend to co-evolve [5, 6] allowing the study of interactions between proteins present in all analyzed species. These studies utilized the Codon Adaptation Index (CAI) [7] as a predictor of protein levels. The CAI infers the optimality weights of the various codons by examining the codon usage of the coding sequences of a group of genes that are assumed to be highly expressed in the species examined, and uses these weights to judge the optimality of any coding sequence in this species with respect to translation. The underlying assumption of this method is that the coding sequences of highly expressed genes are well-adapted to the tRNA pools of the cell, so that their codon usage reflects these pools, and therefore allows for the inference of optimality scores for codons, and consequently for coding sequences. The observation that cellular tRNA pools are highly correlated with the tRNA gene copy numbers in the genome [8], allows for the inference of codon optimality scores more directly, without the need to select a group of highly expressed genes. Indeed, a recent study suggested an alternative index of translational optimality – the tRNA Adaptation Index (tAI) [1], and demonstrated its use for the inference of genome-wide translational selection. As a preliminary step towards the analysis of phenotypic divergence using tAI, we examine in this paper the utility of this index as a predictor of protein expression levels, as well as the functional content of multi-species tAI profiles.

2 Results and Discussion

2.1 tAI as an Indicator of Protein Expression Levels

The tAI predicts the level of adaptation of a coding sequence relative to the cell's tRNA pools. As a first test of the functional power of prediction of this index we examined its correspondence with genome-wide experimentally determined protein levels in *S. cerevisiae* [9]. Using the protein levels of almost 4000 *S. cerevisiae* ORFs we obtain a significant positive correlation ($R=0.63$ using Pearson correlation; $p<1e-363$) between tAI values and the corresponding log-transformed protein levels (Fig. 1A). The same analysis, using a different data set constituting 150 proteins [10], yielded similar results. Comparable, yet lower, correlations were obtained using the related indices CAI [7] ($R=0.58$) and FOP' [11] ($R=0.57$).

Significant correlations have been previously observed between CAI and mRNA levels [12], presumably due the general association of high protein levels with high transcript levels. Indeed, the correlation between the log-transformed genome-wide mRNA [13] and protein [9] levels obtained under similar conditions is highly significant with a Pearson correlation coefficient $R=0.62$ ($p<1e-363$; Fig. 1B). However, transcript levels are generally considered poor indicators of protein levels, as similar mRNA levels may be accompanied by a wide range (up to 20-fold difference) of protein levels, and *vice versa* [14]. Some of the discrepancy between mRNA and protein levels may be perhaps explained by different levels of translational control exerted on genes with a similar mRNA level. Such translational control may be manifested in the adaptation of the coding sequence of genes to the tRNA pools of the cell. Therefore, the tAI could potentially provide complementary information to mRNA levels when predicting protein levels. To examine the contribution of the tAI in the prediction of

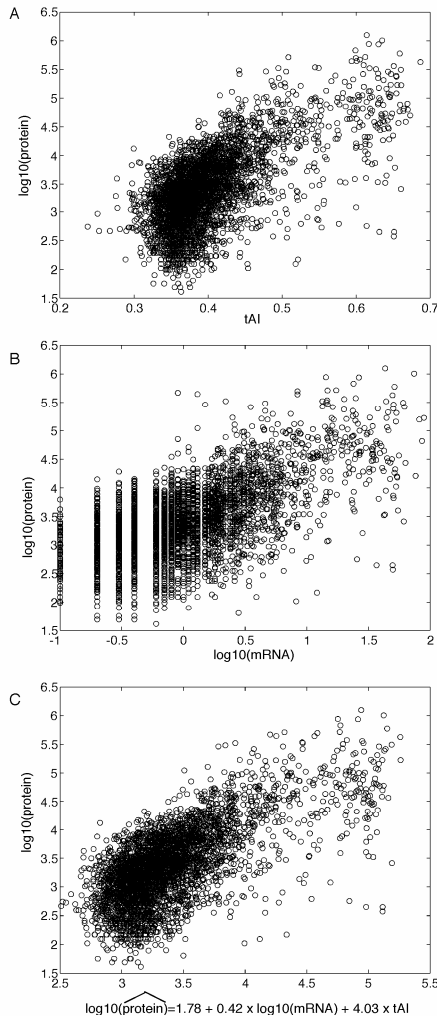


Fig. 1. The relationship of tAI and experimentally-determined mRNA levels with experimentally-determined protein levels in *S. cerevisiae*. A. log-transformed protein levels vs. tAI; B. log-transformed protein levels vs. log-transformed mRNA levels. C. experimentally-determined log-transformed protein levels vs. predicted log-transformed protein levels, obtained from multiple linear regression using both tAI and mRNA levels. mRNA and protein data were obtained from [13] and [9], respectively.

protein levels when mRNA levels are also available we computed a multiple linear regression model utilizing both tAI and log-transformed mRNA levels [13] to predict the log-transformed protein levels [9] (Fig. 1C). The model’s improvement over the individual predictors seems quite modest, with the Pearson correlation coefficient of the fitted values with the log-transformed protein levels being 0.67 ($p < 1e-363$). However, computation of the partial correlations indicates that each of the individual

variables makes a significant contribution: the partial correlation of tAI with the log-transformed protein levels, given the log-transformed mRNA levels is $R=0.34$ ($p=3.44e-100$); and a partial correlation of $R=0.29$ ($p=6.4e-74$) for the log-transformed mRNA and protein levels, given the tAI.

Examination of the scatter-plot of tAI vs. the log-transformed protein levels, reveals that although the correlation between these two variables is highly significant, similar to the observation for mRNA levels, tAI is not a good predictor of protein levels. A possible explanation for the inaccuracies in the predictions of the tAI is that, whereas protein and transcript levels vary across different conditions, the tAI is independent of conditions. Therefore, it is possible that tAI is an indicator of the maximal potential protein levels, rather than the protein levels at a specific condition. To examine the validity of this hypothesis we capitalized on the many microarray experiments of recent years, as compared to few proteomic studies. The fact that high mRNA levels generally correspond to high protein levels, allows us to make a comparison of tAI with mRNA levels rather than protein levels. We examined 24 outliers for the correlation between the tAI and mRNA levels, that exhibited relatively high tAI (greater than 0.5), but lower levels of transcript than would be expected, and looked for experiments where these outliers were induced, using a wild-type yeast strain. This analysis was obviously limited to the conditions covered by experiments published to date, and therefore does not necessarily cover all the conditions yeast cells may experience. However, despite this limitation, in the majority of cases (21/24 cases) we could find a condition under which the ORF was at least twofold induced, with the lowest maximal induction being 3.4-fold for YLR461W, a member of the seripauperin family, during the unfolded protein response [15]. In many cases we could find an experiment for which the product of the mRNA levels at log-phase [13] and the fold-induction value was in line with the expected mRNA level. For example, YPL240C, a cytoplasmic chaperone of the HSP90 family with a tAI value of 0.60, but very modest transcript levels under log-phase growth [13], is induced 11.7-fold during a heat shock experiment from 21°C to 37°C [16]. Thus, available data support our hypothesis of tAI as an indicator of maximal protein levels under all possible conditions encountered by the cell.

2.2 tAI as a Predictor for Translational Selection in a Genome

The application of the tAI to the sequences of a genome is useful only if translational selection has played a significant part in shaping the codon usage of a genome. Thus, before selecting species for a multi-species analysis we checked whether translational selection can be detected in their genome. The effective number of codons (N_c) is a measure of the departure of codon usage in a sequence from random usage of synonymous codons, and is related to the amount of entropy in codon usage [17]. N_c reaches its maximal value (61) when codon usage is completely random, and its minimal value when only one codon is used per amino acid. Therefore, if translational selection were the only force shaping codon usage, sequences selected for optimal translation would be detected by their low values of N_c . However, codon usage, and with it N_c , is largely affected by the silent GC content (X_g), *i.e.* the percentage of codons that have guanine or cytosine at their third nucleotide position. dos Reis et al. [1] have suggested testing for the presence of translational selection in a genome, by assessing the correlation between the tAI, and the difference between $f(X_g)$, a

function predicting Nc based solely on Xg, and Nc. A strong positive correlation in this test would indicate co-adaptation between codon usage and tRNA gene copy numbers. We applied this test to eight ascomycotic yeast species, and found translational selection to be present in all of them (Table 1; Fig. 2). In spite of this, it may be that while translational selection shaped the residual codon bias in coding sequences left after accounting for the effect of silent GC, mutation pressure has been so strong that the effect of translational selection on the overall codon bias in the sequence might be minute. In such a case, changes in protein levels may be achieved in different ways, for example by raising the levels of transcript. This suggests that the test by dos Reis et al is not appropriate for testing the extent of the effect of translational selection in shaping codon usage, and as a consequence on expression. Therefore, to test the contribution of translational selection to overall codon usage, we tested the correlation between tAI and Nc. This time we expect strong negative correlation if codon usage is highly adapted to the cellular tRNA pools. We found that for seven of the species this correlation was highly significant (Table 1; Fig. 3A). However, for *A. gossypii* the magnitude of correlation was low and insignificant (Table 1; Fig. 3B), suggesting that for this species tAI would not be a good predictor of expression levels. We therefore excluded *A. gossypii* from the subsequent analysis.

Table 1. Pearson correlation of tAI with $f_1(X_g)$ -Nc and with Nc in various ascomycotic yeast species

species	correlation of tAI with		correlation of	
	$f_1(X_g)$ -Nc	significance	tAI with Nc	significance
<i>A. gossypii</i>	0.60	< 0.001	-0.38	0.384
<i>C. albicans</i>	0.62	0.002	-0.65	0.005
<i>C. glabrata</i>	0.86	<0.001	-0.79	<0.001
<i>D. hansenii</i>	0.78	<0.001	-0.75	<0.001
<i>S. bayanus</i>	0.81	<0.001	-0.73	<0.001
<i>S. cerevisiae</i>	0.81	<0.001	-0.79	<0.001
<i>S. pombe</i>	0.83	<0.001	-0.66	<0.001
<i>Y. lipolytica</i>	0.82	<0.001	-0.84	<0.001

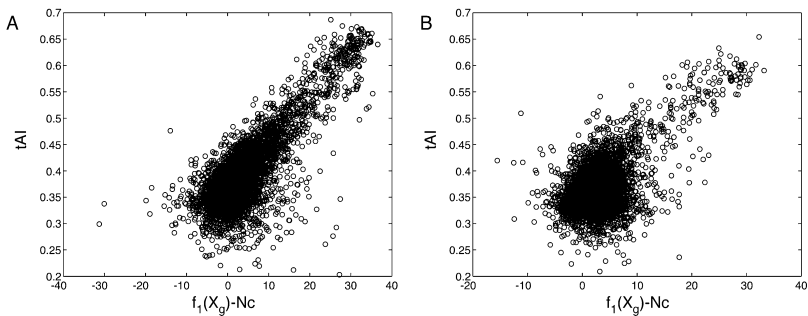


Fig. 2. tAI vs. $f_1(X_g)$ -Nc for *S. cerevisiae* (A) and *A. gossypii* (B) ORFs

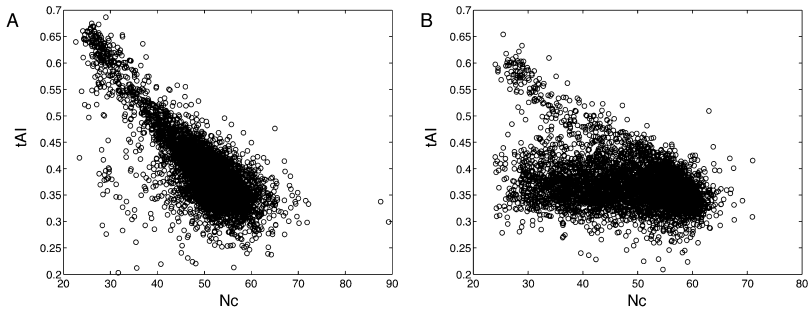


Fig. 3. tAI vs. Nc for *S. cerevisiae* (A) and *A. gossypii* (B) ORFs

In general, we can conclude that the tAI cannot be used as a predictor of protein expression levels in all genomes. It has already been suggested that translational selection may not be operating in all genomes [1]. However, our results indicate that even in cases where translational selection can be shown to be present, as indicated by the difference between $f(Xg)$ and Nc, it may still be that translational selection makes an insignificant contribution to the overall codon bias, making the tAI an inappropriate predictor of protein expression levels.

2.3 Multi-species tAI Profiles Contain Functional Information

If phenotypic divergence between species were the product of different expression regimes, it is expected that the levels of the proteins underlying the phenotype would vary across species in a coordinated manner under the relevant conditions. In this respect the prediction of protein levels from coding sequences seems problematic, since these predictions are condition-independent. Yet, recent studies [5, 6], using CAI as a predictor for protein expression levels, showed that the profiles of predicted expression levels across species tended to be correlated for functionally interacting protein pairs, indicating that functional inferences based on predicted expression levels might be possible.

To validate the use of tAI for functional inferences we checked the behavior of tAI profiles of genes across species for a set of experimentally-determined interacting protein pairs taken from the data of [18]. We generated, using sequence similarity measures, a table of close to 5000 orthologous groups in the seven ascomycotic yeast species that showed a significant influence of translational selection over their codon usage patterns (Table 1). Over 2500 of these groups were present in at least six of the eight species, including *S. cerevisiae* and *S. pombe*, and were used for further analysis. For each orthologous group we generated a profile of predicted expression levels across species, using the tAI. We thus obtained a matrix where each column corresponds to a yeast species, and each row to an orthologous group. We compared the distribution of Pearson correlation coefficients found among the tAI profiles of protein pairs known to interact, with those of two sets of non-interacting protein pairs. The first set of non-interacting pairs, C, was obtained by calculating all possible pairs using the proteins in the interacting pairs set, and then subtracting those pairs that are known to interact. The second control, C', was a random sample of 1311 pairs from C, the same number of pairs as in the set of known physically interacting pairs. We found the correlations among the physically interacting pairs to be significantly higher

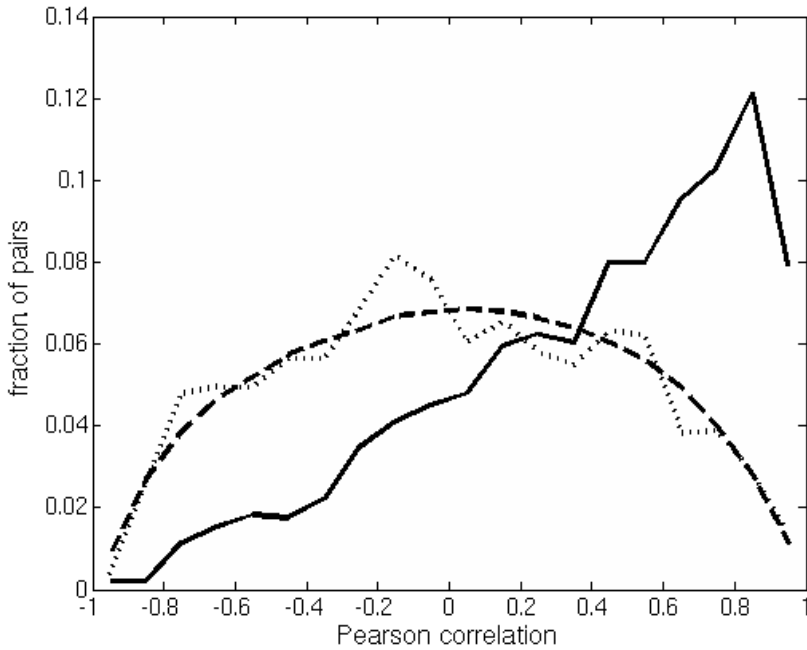


Fig. 4. Distribution of Pearson correlation coefficients among the tAI profiles of protein pairs known to physically interact (solid line), as compared to the corresponding distributions of the two sets of non-interacting proteins C (dashed line) and C' (dotted line)

than those found in both sets of non-interacting pairs ($p < 1e-100$ for both comparisons using a Wilcoxon-Cox rank sum test; Fig. 4).

3 Conclusion

The tAI, an intuitive measure of the optimality of a coding sequence in terms of translation, correlates well with experimentally-determined protein and mRNA levels, and may be a good predictor of the maximal levels of protein under all conditions encountered by a species. The putative levels of protein predicted by the tAI tend to vary in a coordinated manner across species for physically interacting pairs, indicating the potential of this index to serve in functional inferences regarding phenotypic differences among closely-related species. However, care should be taken to apply the tAI only to genomes where translational selection can be shown to be a major force shaping codon usage in sequences.

4 Data and Methods

4.1 Species Analyzed

The yeast species used in this study are *Saccharomyces cerevisiae*, *Saccharomyces bayanus*, *Candida glabrata*, *Ashbya gossypii*, *Debaryomyces hansenii*, *Candida albicans*, *Yarrowia lipolytica* and *Schizosaccharomyces pombe*.

4.2 Protein and Coding Sequences

C. albicans protein and coding sequences were downloaded from [19]. *S. cerevisiae* and *S. bayanus* protein and coding sequences were downloaded from [20]. For *S. bayanus* several sequences may correspond to different fragments of the same ORF. We used the annotation given by [21] to merge such fragments.

For the remaining five species files in both fasta and UniProt formats were downloaded from [22]. The UniProt format files were used to construct a dictionary, linking accessions referring to nucleotide sequences to their corresponding proteins. We then downloaded, from [23] all entries corresponding to the species in question and containing a “CDS” feature, in EMBL format. A perl script utilizing BioPerl [24] was then used to go over the EMBL format file to extract coding sequences of accessions corresponding, according to the dictionary, to sequences in the protein fasta file. Coding sequences were used only if their length was at least three times the length of the protein sequence. If the coding sequence was longer than this length we assumed that this was due to an alternative initiation of the sequence, and used the last N nucleotides, where N is the expected length for the coding sequence.

4.3 tRNA Gene Copy Numbers

For all species except *C. albicans* and *S. bayanus* the tRNA gene copy numbers were obtained by applying the tRNAscan-SE software version 1.1 [25] to chromosome sequences obtained from GenBank (<http://www.ncbi.nlm.nih.gov/Genbank>). For *S. bayanus* we used the tRNA gene copy numbers of the closely related *S. cerevisiae*. Although a list of the tRNA genes for this species is available [21], the low total number of protein-coding genes available for it and the other two *sensu stricto* species sequenced in the same project (less than 5000 in each of the three species, compared to close to 6000 in *S. cerevisiae*), indicates that the quality of the genome sequence may not be high enough to reliably determine the copy numbers of tRNA genes. The strong conservation of synteny between *S. bayanus* and *S. cerevisiae* [21] and the relatively short time that has passed since their divergence (~20 million years ago) makes the use of the tRNA gene copy numbers from *S. cerevisiae* a conservative choice.

For *C. albicans* we extracted the tRNA gene counts from [19].

4.4 Calculation of tAI, Nc, $f_1(X_2)$ -Nc, Their Correlation and Its Significance for Coding Sequences

The tAI method is described in detail in [1]. Briefly, the method entails calculating a weight for each of the sense codons, derived from the copy numbers of all the tRNA types that recognize it. For a given coding sequence, the tAI value is then the geometric mean of the weights of all its sense codons (stop codons were ignored when encountered). To calculate the tAI for coding sequences we used the codonR scripts supplied by [1], downloaded from <http://people.cryst.bbk.ac.uk/~fdosr01/tAI/>, which we modified to include the first codon, as well as other methionines. The effective number of codons (Nc,[17]) was calculated with the modified version of the codonW program, supplied by [1], which was downloaded from the same site. This version of codonW was further modified to accommodate the alternative yeast nuclear code used

by *D. hansenii* [26] and *C. albicans* [27]. The significance of the observed correlation of Nc with tAI was calculated by permuting the tAI weights of the sense codons 1000 times. Each such permutation was then used to compute the correlation of Nc with the tAI calculated using the randomized weights. These calculations were done using scripts downloaded from the same site and the R software for statistical computing (<http://www.r-project.org>).

4.5 Generation of a Table of Orthologous Groups

The orthologous groups were constructed using a *S. cerevisiae*-centered methodology. We constructed, using the inparanoid algorithm [28], six lists of orthologous groups containing genes from only two species – *S. cerevisiae* and one of the other six species. These two-way groups were then merged to obtain orthologous groups potentially encompassing all species in the sample.

The generation of the lists of two-species orthologous groups utilized the inparanoid algorithm [28] without an outgroup and without bootstrapping. We kept only sequences that were assigned a confidence value of at least 25% for their membership in the group. There is a discrepancy between the inparanoid algorithm as reported in [28], and the program supplied by the authors at <http://inparanoid.cgb.ki.se/>: while the paper specifies that the matched segment between two sequences must cover at least 50% of the longer sequence in order for the sequences to be considered homologous, the program applies this cutoff to the shorter sequence. In order to avoid domain-level matches, we modified the inparanoid program to reflect the algorithm as presented in the paper.

The second part, the merger of the six two-species lists into one seven-species list, was achieved by iteratively adding the two-species lists. The order of processing of the lists was dictated by the relative closeness of the second species in the list to *S. cerevisiae* (*S. cerevisiae* was included in all lists): starting with *S. cerevisiae*'s most distant relative (*S. pombe*) and ending with its closest relative (*S. bayanus*). *C. albicans* and *D. hansenii* are equidistant from *S. cerevisiae*, and we arbitrarily chose to first analyze the *C. albicans* list. Each iteration consisted of going over all orthologous groups in the list being processed. For each such group, if its *S. cerevisiae* genes were found in a group obtained in a previous iteration, the two groups were merged; otherwise, if the genes in the group appeared after the divergence of *S. cerevisiae* from the previously analyzed species, a new group was created. Note that if a duplication had occurred after the divergence from the previously analyzed species then more than one group may be merged with the same pre-existing group. The order of merger, *i.e.* according to the order of divergence from *S. cerevisiae*, ensures that there will be no ambiguity as to which pre-existing group to add a currently analyzed group.

4.6 Generation of a Matrix of Predicted Expression Levels Across Species

We combined the orthologous groups table with tAI scores to create a matrix of predicted expression levels across species. In cases where the orthologous group contained several paralogs we used the maximal tAI score among them. We discarded all profiles that had no representative from *S. pombe*. The resultant matrix of predicted

expression was then submitted for preprocessing at the GEPAS server [29]: profiles with more than 30% missing values were removed, missing values in the remaining profiles were imputed using the KNNimpute algorithm with $k=15$. This left us with 2592 profiles. Each column of the matrix was subsequently normalized, so that for each species the mean and standard deviation would be 0 and 1, respectively.

4.7 Analysis of Outliers Having High tAI, But Relatively Low mRNA Expression Levels

We analyzed two groups of outliers in the comparison of tAI and experimentally determined mRNA levels obtained from [13]: ORFs with $0.6 \leq \text{tAI}$ and transcript levels at most 10 (9 ORFs), and ORFs with $0.5 \leq \text{tAI} < 0.6$ and transcript levels at most 1 (20 ORFs). Since mRNA levels are subject to noise, results for the same condition may vary across experiments, and what may seem as an outlier using the data of one experiment may not be an outlier using the data of another experiment. We therefore filtered out from the set of outliers five ORFs that were not outliers using another dataset obtained under similar conditions (dataset GSM6711 corresponding to one of the control samples in the study of [30]). The control dataset was downloaded from the GEO database [31] (<http://www.ncbi.nlm.nih.gov/geo>). We then used the “expression connection” tool at the SGD database site [32] to obtain the maximal induction levels of the outlying ORFs, taking care to consider only experiments in which a wild-type strain was used.

4.8 Pairs of Physically Interacting Genes in *S. crevisiae*

We extracted 2301 pairs of proteins from the supplementary data of von Mering et al [18] by filtering out those protein pairs that were marked as “previously annotated: no”. We further filtered out pairs of paralogs belonging to the same orthologous groups, and pairs where at least one member was not in the data matrix, leaving 1311 pairs of interacting proteins.

References

1. dos Reis, M., Savva, R., Wernisch, L.: Solving the riddle of codon usage preferences: a test for translational selection. *Nucleic Acids Res* **32** (2004) 5036-5044
2. Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., Yeates, T.O.: Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A* **96** (1999) 4285-4288
3. Avidor-Reiss, T., Maer, A.M., Koundakjian, E., Polyanovsky, A., Keil, T., Subramaniam, S., Zuker, C.S.: Decoding cilia function: defining specialized genes required for compartmentalized cilia biogenesis. *Cell* **117** (2004) 527-539
4. Martin, M.J., Herrero, J., Mateos, A., Dopazo, J.: Comparing bacterial genomes through conservation profiles. *Genome Res* **13** (2003) 991-998
5. Fraser, H.B., Hirsh, A.E., Wall, D.P., Eisen, M.B.: Coevolution of gene expression among interacting proteins. *Proc Natl Acad Sci U S A* **101** (2004) 9033-9038
6. Lithwick, G., Margalit, H.: Relative predicted protein levels of functionally associated proteins are conserved across organisms. *Nucleic Acids Res* **33** (2005) 1051-1057

7. Sharp, P.M., Li, W.H.: The codon Adaptation Index--a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res* **15** (1987) 1281-1295
8. Percudani, R., Pavesi, A., Ottonello, S.: Transfer RNA gene redundancy and translational selection in *Saccharomyces cerevisiae*. *J Mol Biol* **268** (1997) 322-330
9. Ghaemmaghami, S., Huh, W.K., Bower, K., Howson, R.W., Belle, A., Dephoure, N., O'Shea, E.K., Weissman, J.S.: Global analysis of protein expression in yeast. *Nature* **425** (2003) 737-741
10. Greenbaum, D., Jansen, R., Gerstein, M.: Analysis of mRNA expression and protein abundance data: an approach for the comparison of the enrichment of features in the cellular population of proteins and transcripts. *Bioinformatics* **18** (2002) 585-596
11. Lavner, Y., Kotlar, D.: Codon bias as a factor in regulating expression via translation rate in the human genome. *Gene* **345** (2005) 127-138
12. Friberg, M., von Rohr, P., Gonnet, G.: Limitations of codon adaptation index and other coding DNA-based features for prediction of protein expression in *Saccharomyces cerevisiae*. *Yeast* **21** (2004) 1083-1093
13. Holstege, F.C., Jennings, E.G., Wyrick, J.J., Lee, T.I., Hengartner, C.J., Green, M.R., Golub, T.R., Lander, E.S., Young, R.A.: Dissecting the regulatory circuitry of a eukaryotic genome. *Cell* **95** (1998) 717-728
14. Gygi, S.P., Rochon, Y., Franza, B.R., Aebersold, R.: Correlation between protein and mRNA abundance in yeast. *Mol Cell Biol* **19** (1999) 1720-1730
15. Travers, K.J., Patil, C.K., Wodicka, L., Lockhart, D.J., Weissman, J.S., Walter, P.: Functional and genomic analyses reveal an essential coordination between the unfolded protein response and ER-associated degradation. *Cell* **101** (2000) 249-258
16. Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O.: Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell* **11** (2000) 4241-4257
17. Wright, F.: The 'effective number of codons' used in a gene. *Gene* **87** (1990) 23-29
18. 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** (2002) 399-403
19. Arnaud, M.B., Costanzo, M.C., Skrzypek, M.S., Binkley, G., Lane, C., Miyasato, S.R., Sherlock, G.: "Candida Genome Database" <http://www.candidagenome.org/>. (14 August 2005)
20. Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dolinski, K., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Hong, E.L., Nash, R., Oughtred, R., Skrzypek, M., Theesfeld, C.L., Binkley, G., Dong, Q., Lane, C., Sethuraman, A., Weng, S., Botstein, D., Cherry, J.M.: "Saccharomyces Genome Database" <ftp://ftp.yeastgenome.org/yeast/>. (16 June 2005)
21. Kellis, M., Birren, B.W., Lander, E.S.: Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature* **428** (2004) 617-624
22. Pruess, M., Kersey, P., Apweiler, R.: The Integr8 project--a resource for genomic and proteomic data. *In Silico Biol* **5** (2005) 179-185
23. Kanz, C., Aldebert, P., Althorpe, N., Baker, W., Baldwin, A., Bates, K., Browne, P., van den Broek, A., Castro, M., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Gamble, J., Diez, F.G., Harte, N., Kulikova, T., Lin, Q., Lombard, V., Lopez, R., Mancuso, R., McHale, M., Nardone, F., Silventoinen, V., Sobhany, S., Stoehr, P., Tuli, M.A., Tzouvara, K., Vaughan, R., Wu, D., Zhu, W., Apweiler, R.: The EMBL Nucleotide Sequence Database. *Nucleic Acids Res* **33** (2005) D29-33

24. Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G., Korf, I., Lapp, H., Lehtvaslaiho, H., Matsalla, C., Mungall, C.J., Osborne, B.I., Pockock, M.R., Schattner, P., Senger, M., Stein, L.D., Stupka, E., Wilkinson, M.D., Birney, E.: The Bioperl toolkit: Perl modules for the life sciences. *Genome Res* **12** (2002) 1611-1618
25. Lowe, T.M., Eddy, S.R.: tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res* **25** (1997) 955-964
26. Tekaa, F., Blandin, G., Malpertuy, A., Llorente, B., Durrens, P., Toffano-Nioche, C., Ozier-Kalogeropoulos, O., Bon, E., Gaillardin, C., Aigle, M., Bolotin-Fukuhara, M., Casaregola, S., de Montigny, J., Lepingle, A., Neuveglise, C., Potier, S., Souciet, J., Wesolowski-Louvel, M., Dujon, B.: Genomic exploration of the hemiascomycetous yeasts: 3. Methods and strategies used for sequence analysis and annotation. *FEBS Lett* **487** (2000) 17-30
27. Sugita, T., Nakase, T.: Non-universal usage of the leucine CUG codon and the molecular phylogeny of the genus *Candida*. *Syst Appl Microbiol* **22** (1999) 79-86
28. Remm, M., Storm, C.E., Sonnhammer, E.L.: Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J Mol Biol* **314** (2001) 1041-1052
29. Herrero, J., Al-Shahrour, F., Diaz-Uriarte, R., Mateos, A., Vaquerizas, J.M., Santoyo, J., Dopazo, J.: GEPAS: A web-based resource for microarray gene expression data analysis. *Nucleic Acids Res* **31** (2003) 3461-3467
30. Bulik, D.A., Olczak, M., Lucero, H.A., Osmond, B.C., Robbins, P.W., Specht, C.A.: Chitin synthesis in *Saccharomyces cerevisiae* in response to supplementation of growth medium with glucosamine and cell wall stress. *Eukaryot Cell* **2** (2003) 886-900
31. Barrett, T., Suzek, T.O., Troup, D.B., Wilhite, S.E., Ngau, W.C., Ledoux, P., Rudnev, D., Lash, A.E., Fujibuchi, W., Edgar, R.: NCBI GEO: mining millions of expression profiles--database and tools. *Nucleic Acids Res* **33** (2005) D562-566
32. Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dolinski, K., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Hong, E.L., Nash, R., Oughtred, R., Skrzypek, M., Theesfeld, C.L., Binkley, G., Dong, Q., Lane, C., Sethuraman, A., Weng, S., Botstein, D., Cherry, J.M.: "Saccharomyces Genome Database" <http://www.yeastgenome.org/>. (1 February 2006)

Improved Duplication Models for Proteome Network Evolution

Gürkan Bebek¹, Petra Berenbrink², Colin Cooper³, Tom Friedetzky⁴,
Joseph H. Nadeau⁵, and S. Cenk Sahinalp^{2,*}

¹ Department of EECS, Case Western Reserve University, Cleveland,
OH 44106-7071 USA

² School of Computing Science, Simon Fraser University, Burnaby BC,
V5A 1S6 Canada

³ Department of Computer Science, King's College, London WC2R 2LS, UK

⁴ Department of Computer Science, Durham University, Durham, DH1 3LE, UK

⁵ Genetics Department, Case Western Reserve University, Cleveland,
OH 44106-4955 USA

Abstract. Protein-protein interaction networks, particularly that of the yeast *S. Cerevisiae*, have recently been studied extensively. These networks seem to satisfy the small world property and their (1-hop) degree distribution seems to form a power law. More recently, a number of duplication based random graph models have been proposed with the aim of emulating the evolution of protein-protein interaction networks and satisfying these two graph theoretical properties. In this paper, we show that the proposed model of Pastor-Satorras et al. does not generate the power law degree distribution with exponential cutoff as claimed and the more restrictive model by Chung et al. cannot be interpreted unconditionally. It is possible to slightly modify these models to ensure that they generate a power law degree distribution. However, even after this modification, the more general k-hop degree distribution achieved by these models, for $k > 1$, are very different from that of the yeast proteome network. We address this problem by introducing a new network growth model that takes into account the sequence similarity between pairs of proteins (as a binary relationship) as well as their interactions. The new model captures not only the k-hop degree distribution of the yeast protein interaction network for all $k > 0$, but it also captures the 1-hop degree distribution of the sequence similarity network, which again seems to form a power law.

1 Introduction

Protein-protein interactions play a central role in the execution of key biological functions of a cell. Such a relationship can be summarized in a *graph* (network) in which each *node* represents a protein and each (undirected) *edge* represents

* Corresponding author: cenk@cs.sfu.ca

an interaction. A graph including *all* proteins in an organism and all possible interactions between these proteins can be called the *proteome network* of that organism.

The structure of the yeast proteome network seems to reveal two interesting graph theoretic properties [20,35]: (i) The degree distribution of nodes (i.e. the proportion of nodes with degree k as a function of degree) approximates a *power-law* (i.e. is approximately ck^{-b} for some constants c, b). (ii) The graph exhibits the *small world effect*.

Small world phenomena and the power-law degree distributions have previously been observed in a number of naturally occurring graphs such as communication networks [14], web graphs [1,4,9,11,21,22], research citation networks [29], human language graphs [15], neural nets [36] etc. These two properties can not be observed in the classical random graph models studied by Erdős and Rényi [13] in which edges between pairs of nodes are determined independently. However, it is possible to generate graphs that satisfy these properties by an iterative process that adds one new node to the graph at each step [1,2,6,8,9,11,21]. The new node is then connected to some b (b can be a constant or an independent random variable) of the existing nodes, each of which is chosen with probability proportional to its degree. Unfortunately such a *preferential attachment* model does not capture the essence of the genome evolution and hence can not be used to model proteome networks. According to Ohno’s model [25], the two underlying mechanisms for genome evolution is gene duplication and point mutations.¹ Recent work, thus, has focused on random graph models that grow via node duplications and get modified by mechanisms that emulate point mutations.

Among these studies, the most promising one, which we call the *general duplication model*, was described independently in [26,34,7]. The general duplication model works in iterations; in each iteration t , one existing node (representing a gene or an associated protein) is chosen uniformly at random and is “duplicated” with all its edges. After the *duplication* step, to emulate mutations, also named as the *divergence* step, each edge of the new node is deleted with probability q . This is followed by inserting edges between the new node and every other node with probability r/t where t is the total number of nodes and r is a constant. With the right selection of parameters q and r , the general duplication model well approximates the degree distribution of the yeast proteome network.

The first serious study to formally analyze the degree distribution of the general duplication model was by Pastor-Satorras et al. who, in [26], claim that the distribution of both the general yeast proteome network and the duplication model is a “power law with exponential cut-off”. This means that the fraction of

¹ After a gene duplication event, one of the genes may accumulate deleterious mutations and be lost, or both copies of the gene may be retained. Two possible evolutionary reasons for keeping both copies can be (i) selection for increased levels of expression, or (ii) divergence of gene function [23,30]. Functional divergence can be produced through complementary degeneration [16]. Although the duplicated regions of the genomes have been described and listed before (for instance *S. Cerevisiae* [31,37]), there is no certain schema of how duplications formed the current shape of the genomes.

nodes with degree k among all nodes is independent of time and is approximated by $f_k = ck^{-b} \cdot a^{-k}$; here a, b, c are constants. However, they make a number of simplifying assumptions in their analysis to get this result. For instance, they approximate the probability for generating a node with degree k by the probability of duplicating a node with degree $k + 1$ only and subsequently deleting a single edge. This assumption also reduces the number of *singletons*. They further approximate this probability with a function linear in k .

A more recent analysis of the degree distribution of the general duplication model, for the special case that $r = 0$ is given by Chung et al. [10]. As per [10], we will refer to this special case as the *pure duplication model*. In contrast to [26], Chung et al. claim that the fraction of nodes with degree k is independent of time and is of the form $f_k = ck^{-b}$; here b is a function of $p = 1 - q$ and values of $b \leq 2$ are possible for some p . The pure duplication model creates *singleton* nodes, i. e. nodes that are not connected to any other node of the graph. Since, a node can only get a new edge if one of its neighbors is copied, a singleton will remain singleton during the whole graph generation process. Note that in this model all non-singleton nodes form one connected component.

In a separate work, van Noort et al. [24] show that the gene coexpression network in *S. Cerevisiae* have scale-free and small-world network properties. By using the homology relations between the genes in coexpression network, they present a model which can generate networks with similar scale-free and small-world properties. The model starts with a number of genes which have a number of transcription factor binding sites (TFBSs) and genes sharing a minimum number of TFBSs considered coexpressed. At every time step each gene can be duplicated or deleted with certain probabilities. Also, at every time step a TFBS of a gene can be deleted or a new TFBS from another gene can be acquired by a gene with certain corresponding probabilities. Different from many, van Noort et al. [24] consider deleting or inserting a TFBS of the gene which deletes a set of connections, or adds a set of links to the gene. Hence, in their approach the connections of genes were considered with groups. Van Noort et al. [24] claim that the model generates a degree distribution with a slope similar to the coexpression network of *S. Cerevisiae*². Additionally, average clustering coefficient³, and shortest path length of the networks were compared. Although these are measures to understand the topology of a network, they are not sufficient to claim that two networks are similar at all.

There is also another study presented by Przulj et al. [28], in which a different approach to model these networks has been studied. Przulj et al. [28] claim that a random geometric model better captures the currently accepted protein

² Numerical results were not presented in [24]. Hence, the simulation results given draws certain amount of question about how close the degree distribution, i.e. the power-law exponent, was.

³ The clustering coefficient of a node is the ratio between the actual number of edges between neighbors of a node and the maximum possible number of edges between these neighbors. Average clustering coefficient of a network is the average of clustering coefficients over all units in the system. [36]

protein interaction networks. A geometric disc graph is formed by connecting two nodes of the graph with an edge, if their distance in the metric space is smaller than a certain threshold. Przulj et al. [28] argue that the scale-free property of the proteomes is a result of the noise in the available data at the moment and the degree distribution of such networks should follow the Poisson distribution. By counting the number of different motifs in the networks, they form a measure of local network structure and using this they compare different models with the available proteomes. According to the experiments they carry out, a three dimensional geometric disc graph with same number of nodes but six times larger edge count has similar number of motifs as the proteomes they worked on. Although, the network motifs considered capture local properties of the networks, in their work, Przulj et al. [28] (i) do not take into account Ohno's Theory [25] which states that, the proteome network should be generated through a process, which attributes the genome sequence growth and evolution to subsequent gene duplications followed by mutations on the gene sequences, (ii) do not consider global properties of the networks before drawing conclusions, such as the average degree or the degree distribution. Moreover, the work presented has vague descriptions on how scale free networks are formed. For instance, there are many models available that can generate scale free networks, but not every scale free network necessarily is generated by emulating proteome network growth i.e. duplication and divergence.

The most recent study that was presented by Ispolatov et al. [18] focuses on duplication-divergence models with completely asymmetric divergence. In a completely asymmetric divergence process, links are removed from the duplicated node only. In their study, Ispolatov et al. examines this model where the evolution is characterized by a single parameter, the link retention probability. They claim that, this single-parameter duplication-divergence network growth model can approximate the degree distribution of real protein-protein interaction networks. Although their model generates similar degree distributions, in reality the network lacks the local structure similarity. For instance, this model would not generate any triangular subgraphs (a clique of three in the network) since the duplication would generate cycles of even length or degree one nodes. However, cycles of any size exists in vast numbers in the real proteome network.

In most of these studies the protein-protein interactions identified by high-throughput yeast two-hybrid screens or inferred from mass spectrometry of coimmunoprecipitated protein complexes were considered. However, analysis based on the agreement of the interaction and expression data show that almost less than half of these interactions are biologically relevant [12]. In a recent study, Han et al. [17] showed that low coverage makes determination of the true topology of the network difficult. Han et al. also showed that sampling the real network through these experiments (since the experiments only reveal partial networks), regardless of the topology of the network that we are looking for, the topology of the sub network that is sampled would have a degree distribution similar to a power law. In other words, according to these experiments, it is not clear whether the proteome network has a power law degree distribution or not. However, in

this paper, we assume that the proteome network should be generated through a process, which attributes the genome sequence growth and evolution to subsequent gene duplications followed by mutations on the gene sequences. Previously, it has been shown that this process would generate a network with power-law degree distribution [26,34,7]. Moreover, we show that the degree distribution of the general duplication model is a power law.

We can summarize our contributions as follows. (i) We show that the degree distribution of the pure duplication model ($r = 0$) cannot be a power law as stated in [10]. (ii) We show that the degree distribution of the general duplication model can not be a power law with exponential cut-off as stated in [26]. In fact, for $r > 0$, it is simply a power law. It is also possible to slightly modify the pure duplication model so that it achieves a power law degree distribution but these details are left to a more complete version of the paper [5] due to space limitations. (iii) The (1-hop) degree distribution of a graph is the distribution of nodes with degree k as a function of k . A more general notion is the ℓ -hop degree distribution which is defined to be the distribution of nodes that can reach k nodes in at most ℓ -hops as a function of k . We observe in this paper that the general duplication model does not capture the ℓ -hop degree distribution of the yeast proteome network for $\ell > 1$. (iv) We describe a new model that takes into account the sequence similarity between protein pairs as a binary relationship in addition to their interactions. Our model accurately captures the ℓ -hop degree distribution of the yeast proteome network for all $\ell > 0$ and yields a good approximation to the degree distribution of the sequence similarity network.

Our specific contributions are as follows. We first show in Section 2 that the (expected) proportion of singletons generated by the pure duplication model ($r = 0$) grows in time. In fact, the only limiting (time independent) solution is $f_0 = 1$ and $f_k = 0$ for all $k > 0$. Note that for the case $p = q = 0.5$ the average degree of nodes in the pure duplication model does not change over time (see Lemma 3). Together with the fact that the fraction of singletons increases in time, this implies that (i) the average degree of non-singletons must increase in time and (ii) there is a single connected component of size $o(t)$ with increasing average degree. It is quite possible that this connected component of the network generated by the pure duplication model exhibits a power law with parameter $b \leq 2$, however this is difficult to establish.

In the rest of Section 2, we show that the degree distribution of the general duplication model (in fact, any random model based on duplications) is not a power law with exponential cut-off as claimed in [26]. We achieve this by showing a bound for the maximum degree of the general duplication model and contrasting it with that of a network which exhibits power law with exponential cut-off.

In [5] we proved that the general duplication model for $r > 0$ and a slightly modified version of the pure duplication model indeed achieve a power law degree distribution as per the yeast proteome network. (Due to space limitations we omit these proofs.) However, a more general measure for capturing the

topological properties of a network is the ℓ -hop degree distribution for all $\ell > 0$. Under this measure (for $\ell > 1$), we show that the (modified) general duplication model is quite different from the yeast proteome network.

In Section 3, we finally present our *sequence similarity enhanced model* which is based on the observation that the interactions of sequence-wise similar proteins are highly correlated. The model thus employs *sequence similarity edges* between pairs of nodes/proteins to better capture the mechanisms for updating the interactions after a duplication event. Our model not only captures the degree distribution of the yeast proteome network but also yields a much better approximation to its ℓ -hop degree distribution for $\ell > 1$. Moreover we have observed that the average clustering coefficients of networks generated by this model and the original proteome network are almost equal to each other.

1.1 Preliminaries

We first define the general duplication model formally. The general duplication model grows iteratively in discrete time steps. Let $G(t-1)$ be the network at the end of time step $t-1$. In time step t exactly one new node is generated and will be denoted as v_t . For any node v_s , we will denote its degree (or expected degree if the context is clear) at time step $t \geq s$ by $d_s(t)$.

(i) At each time step t , the new node v_t is generated by picking one of the nodes w in $G(t-1)$ uniformly at random and “duplicating” it to create v_t ; i. e. v_t will initially be connected to all neighbors of w .

(ii) The edges incident to v_t are updated through the following random process. Each edge e is considered independently and is deleted with probability q ($= 1-p$). Then, each node u which is not connected to v_t is considered independently and an edge between u and v_t is created with probability r/t . As mentioned earlier, when $r = 0$ we have the pure duplication model; we show in the next section that it can not achieve a power law degree distribution as stated in [10]. To address this problem the pure duplication model can be modified via a new step (3) where v_t is connected to a uniformly chosen random node (either at all times or only if it had become a singleton at the end of step (2)). As a result, v_t never has degree 0.

Let $\mathbf{F}_k(t)$ denote the number of nodes of degree k at the end of step t in the random process and let $\mathbf{F}(t) = (\mathbf{F}_0(t), \mathbf{F}_1(t), \dots)$ be the degree sequence. Also let $F_k(t) = \mathbf{E}\mathbf{F}_k(t)$ be the expected value, and $f_k(t) = F_k(t)/t$ the expected fraction of nodes of degree k . Finally let $\mathbf{e}(t)$ be the number of edges in $G(t)$ and $e(t) = \mathbf{E}\mathbf{e}(t)$; similarly let $\mathbf{h}(t)$ be the average degree of a node (averaged over all nodes) in $G(t)$, and $h(t) = \mathbf{E}\mathbf{h}(t)$. We say a model has a power law degree sequence if we can find $b, c > 0$ constant such that $f_k(t) \rightarrow f_k$ as $t \rightarrow \infty$ where $f_k = (1 + O(1/k))ck^{-b}$.

2 On the General Duplication Model

This section is on the previous studies on the analysis of the general duplication model. We first show in Section 2.1 that the fraction of singletons in the pure

duplication model grows with time in such a way that $F_0(t) \rightarrow t$ is the only consistent limiting solution. This implies that, unless $f_k = 0$ for $k \geq 1$ then $F_k(t) \neq tf_k$, where f_k is a time independent solution for the limiting proportion of nodes of degree k . In fact, for the particularly interesting case that $p = q = 1/2$, we show that the expected number of non singletons at time step t is between $O(\sqrt{t})$ and $O(t/\log \log t)$. This contradicts the assumption in Eqn(6) of [10]. Thus, without some modification, the pure duplication model of [10] cannot have a power law degree distribution in the form $F_k(t) \sim ctk^{-b}$ for any constants c, b .

Section 2.2 is on the analysis in [26] which predicts the general duplication model to have a degree distribution of the form ‘power law with exponential cut-off’; i. e. there exists constants a, b, c such that, as $t \rightarrow \infty$, we have $f_k(t) \sim ck^{-b}a^{-k}$ for $k \rightarrow \infty$. We show that this cannot be true by demonstrating that the expected maximum degree for a power law with exponential cut-off is $O(\log t)$ whereas the general duplication model has expected maximum degree of $\Omega(tp)$.

2.1 Properties of the Pure Duplication Model

Lemma 1. *The expected proportion of singletons, $f_0(t)$, in the pure duplication model is a non-decreasing function of t and tends to a limit $f_0 \leq 1$. If also we have that $f_k(t) \rightarrow f_k$ for $k \geq 1$ then $f_0 = 1$ and $f_k = 0$ for $k \geq 1$.*

Proof. We have the following recurrence for singletons in the pure duplication model:

$$F_0(t + 1) = F_0(t) + \sum_{k \geq 0} \frac{F_k(t)q^k}{t}.$$

Thus writing $F_k(t) = tf_k(t)$ we have

$$(t + 1)(f_0(t + 1) - f_0(t)) = \sum_{k \geq 1} f_k(t)q^k \geq 0,$$

and we see that $f_0(t + 1) \geq f_0(t)$. As $f_0(t) \leq 1$ it follows that $f_0(t) \rightarrow f_0 \leq 1$ from below as $t \rightarrow \infty$.

Suppose next that for some $k \geq 1$, k constant, $f_k(t) \rightarrow f_k > 0$, then $\sum_{k \geq 1} f_k q^k = c > 0$. Thus there exists T such that for $t \geq T$, $\sum_{k \geq 1} f_k(t)q^k \geq c/2 > 0$ and

$$f_0(t + 1) \geq f_0(t) + \frac{c}{2(t + 1)}.$$

Iterating this we get

$$f_0(t) \geq \frac{c}{2} \log t/T + O(1/T) + f_0(T)$$

i. e. , $f_0(t) > 1$ for t large enough, which is impossible. □

This lemma excludes the existence of power law solutions $f_k \sim ck^{-b}$ for finite $k \geq 1$ (which are suggested in [10]), but we cannot exclude non-limiting degree distributions by this argument.

It is possible to obtain a tighter estimate on the proportion of singletons in the network for the particularly interesting case that $p = q = 1/2$. As per Lemma 3 (see below), this case preserves the (expected) average degree of the nodes throughout the generation of $G(t)$. Thus, $e(t) = e(0) \cdot t$ (where $e(0)$ is the number of edges of $G(0)$).

Lemma 2. *Consider the case $q = 1/2$. Let $F^+(t) = t - F_0(t)$, the number of non-singleton nodes at time t and $F^+ = \mathbf{E}F^+$. Then, there are constants $c_1, c_2 > 0$ such that $c_1\sqrt{t} \leq F^+(t) \leq c_2t/\log \log t$.*

Proof. We have the following recurrence:

$$F^+(t + 1) = F^+(t) + \frac{1}{t} \sum_{k \geq 0} F_k(t)(1 - (1/2)^k) \tag{1}$$

Thus:

$$F^+(t + 1) = F^+(t) + \frac{F^+(t)}{t} - \frac{F^+(t)}{t} \sum_{k \geq 1} \frac{F_k(t)}{F^+(t)} \frac{1}{2^k} \tag{2}$$

As $F_1(t) \leq F^+(t)$, one can easily check $F^+(t) \geq F^+(0)\sqrt{t}$ giving the lower bound.

Now let $g(k) = 1/2^k$, which is convex and thus for any set of λ_k for which $\sum \lambda_k = 1$, we must have $\sum \lambda_k g(k) \geq g(\sum k \lambda_k)$. Now pick $\lambda_k = \frac{F_k(t)}{F^+(t)}$. We have $\sum k F_k(t) = 2e(t) = 2e(0)t$. Thus:

$$\sum_{k \geq 1} \frac{F_k(t)}{F^+(t)} \left(\frac{1}{2}\right)^k \geq \left(\frac{1}{2}\right)^{2e(t)/F^+(t)} \tag{3}$$

By substituting (3) into (2) and using $e(t) = e(0)t$ we get:

$$F^+(t + 1) \leq F^+(t) + \frac{F^+(t)}{t} \left(1 - \left(\frac{1}{2}\right)^{2e(0)t/F^+(t)}\right).$$

This is only satisfied if $F^+(t) \leq c_2t/\log \log t$. This can be verified as follows. Let $c_2 = 4e(0) \log 2$. Either $F^+(t) \leq c_2t/\log \log t$, or if not we can substitute this lower bound into the exponent on the right hand side and iterate the recurrence on t to obtain a contradiction. □

Lemma 3 (below) states that the expected number of edges is $e(t) = ct^{2p}$ and consequently the expected average degree is $h(t) = 2ct^{2p-1}$. Thus for $p < 0.5$ the average degree decreases over time and for $p > 0.5$ it increases. Only for $p = 0.5$ the average degree remains constant; however as the proportion of singletons is $\geq 1 - O\left(\frac{1}{\log \log t}\right)$ due to Lemma 2, the average degree of non-singletons (which all form a single connected component) is $\geq c \log \log t$.

Proposition 1. *The power law exponent b in [10] is given by the solution of $1 = bp - p + p^{b-1}$ and has the value 2 when $p = 1/2$. This is incompatible with $e(t) = 2e(0)t$ unless the connected component is of size $o(t)$.*

To see this, recall that $\sum kF_k(t) = 2e(t)$. Under the assumption that we have a power law degree distribution at $p = 1/2$, then $F_k(t) \sim ck^{-2}t$ and

$$e(t) = \frac{ct}{2} \sum_{k \geq 1} \left(1 + O\left(\frac{1}{k}\right)\right) k^{-1}.$$

However $\sum_{k=1}^{k^*} k^{-1}$ diverges as $k^* \rightarrow \infty$, and we cannot have $e(t) = 2e(0)t$, unless we truncate k^* at a finite value. Lemma 4 (below) sets the expected maximum degree in the pure model at $\Omega(t^p)$, and the power law assumption itself is not compatible with k^* being finite.

It is however still possible that a power law with exponent $b = 2$ holds for the connected component C . Putting $k^* = O(t^{1/2})$ we see that $\sum k^{-1} = O(\log t)$ which gives $e(t) = 2e(0)t$ provided $|C| = O(t/\log t)$, in accordance with the results of Lemma 2.

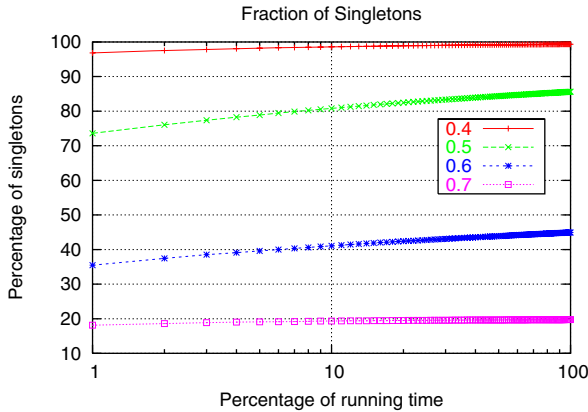


Fig. 1. Percentage of singletons in the pure duplication model as function of time (each curve is for a different value of p)

Lemma 3. *The expected total number of edges and the expected average degree of nodes at step t satisfy*

$$e(t) \sim e(0)t^{2p} \text{ and } h(t) \sim h(0)t^{2p-1}$$

Proof. The number of edges at time $t + 1$ in terms of the number of edges at time t is

$$\mathbf{E}(e(t + 1) \mid e(t)) = e(t) + \frac{1}{t} \sum_{s \leq t} pd_s(t).$$

The first term is trivial; the second term is obtained by considering the possibility that each given node v_s is duplicated at time t ; then $pd_s(t)$ would be the expected number of its edges retained. Because the sum of the degrees of all nodes is twice the number of edges, we have, taking expectations again, that

$$e(t + 1) = \left(1 + 2\frac{p}{t}\right) e(t)$$

which has a solution $e(t) \sim e(0)t^{2p}$. □

Figure 1 shows the percentage of the singletons in the network over the time for different values of p . The model was run until 1000000 non-singleton nodes were created. The plot uses a linear scale on the y-axis (percentage of singletons) and a logarithmic scale on the x-axis (running time).

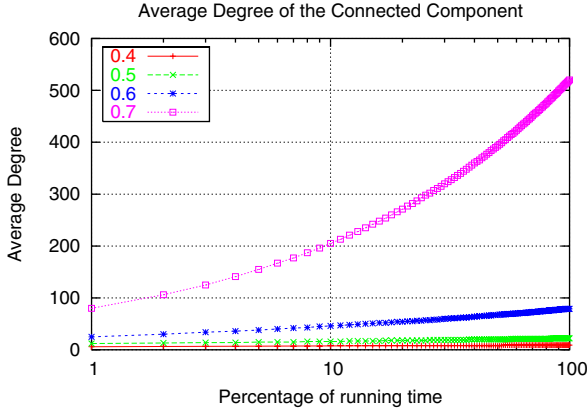


Fig. 2. Average degree of non-singleton nodes in the pure duplication model as function of time (each curve is for a different value of p)

Figure 2 shows the average degree over time for different values of p . Again, the model was run until 1000000 non-singleton nodes were created. The average degree of the network increases by time and the larger the value of p is, the larger is the increases of the average degree.

2.2 On the Degree Distribution of the General Duplication Model

The next lemma shows that the degree distribution of the general duplication model can not be a power law with exponential cut-off as suggested in [26].

Lemma 4. *Let $a, b, c > 0$ be constants. The degree distribution of the general duplication model cannot be in the form $F_k(t) \sim ctk^{-b}a^{-k}$ as claimed in [26].*

Proof. Denote by k_{max} , the expected maximum degree in $G(t)$. Assume an exponential cut-off i. e. $F_k(t) \sim ctk^{-b}a^{-k}$. Then $\sum_{k \geq k_0} F_k(t) = o(1)$ for $k_0 > \log t / \log a$, and so $k_{max} = O(\log t / \log a)$.

On the other hand consider the expected degree of the node v_s at time $t + 1$, which is a non-decreasing function of t . Even in the worst case situation ($r = 0$) we have:

$$d_s(t + 1) = d_s(t) + \frac{d_s(t)}{t}p \tag{4}$$

as the degree of v_s can only increase if one of its neighbors is picked at time t and the edge is retained. Thus:

$$d_s(t + 1) = d_s(t) \left(1 + \frac{p}{t}\right) = d_s(s) \left(1 + \frac{p}{s}\right) \cdot \left(1 + \frac{p}{s+1}\right) \dots \left(1 + \frac{p}{t}\right)$$

Since $\log(1 + x) = x - O(x^2)$ we have

$$\exp\left(\sum_{\tau=s}^t \log(1 + p/\tau)\right) \sim \exp\left(p \sum_{\tau=s}^t 1/\tau\right) = e^{p \log(t/s)}$$

which implies that $d_s(t + 1) = \Omega(d_s(s)(t/s)^p)$ and that $k_{max} = \Omega(t^p)$ contradicting the claim. \square

We finally prove that for $r > 0$ there are no degenerate limiting solutions of the form $f_0 = 1, f_k = 0, k \geq 1$ for the general model of [26].

Lemma 5. *For any $r > 0$ constant, the general model does not have a degenerate limiting solution of the form $f_0 = 1, f_k = 0, k \geq 1$.*

Proof. We have the following recurrence for the expected number of singletons:

$$F_0(t + 1) = F_0(t) + \sum_{k \geq 0} \frac{F_k(t)}{t} q^k \left(1 - \frac{r}{t}\right)^t - \frac{r}{t} F_0(t).$$

Assuming the existence of a limiting solution $F_k(t) = f_k t$ we have (after taking limits):

$$(1 + r - e^{-r}) \cdot f_0 = e^{-r} \sum_{k \geq 1} f_k q^k.$$

If $f_0 = 1$ then $\sum_{k \geq 1} f_k q^k = 0$, but $1 + r - e^{-r} > 0$ for $r > 0$ contradicting this. \square

3 An Enhanced Duplication Model Based on Protein Sequence Similarity

The general duplication model well approximates the degree distribution of the yeast proteome network as observed previously in [26]. (In fact, we have shown in [5] that this degree distribution is simply a power law for $r > 0$; due to space limitations this proof is omitted.) In Figure 3 we compare the degree distribution of the yeast proteome network from the Database of Interacting Proteins (DIP) [38] [4] to that of the (modified) general duplication model with the best fitting [5] parameters $p = 0.465$ and $r = 0.08$. Although the DIP database is incomplete and includes several interactions which are not commonly observed, it still

⁴ The DIP yeast data has ≈ 15000 interactions among 6700 known yeast proteins. The DIP network has only 4700 of the proteins present in the network, which also means that there are about 2000 singletons in the network.

⁵ For all plots, the fits were achieved by calculating the average slope in both curves.

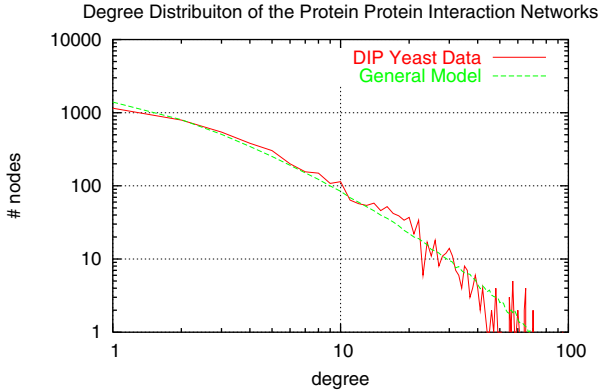


Fig. 3. The degree distribution of the proteome interaction network of the yeast and that of the general model with parameters $q = 0.535$, $r = 0.08$

provides the most comprehensive protein-protein interaction data for the yeast *S.Cerevisiae*. As observed earlier, the degree distribution of the yeast proteome network is very similar to that of the general model with the above parameters.

The degree distribution is one possible measure for testing the structural similarity of two networks. Unfortunately structurally very different networks can have identical degree distributions. For example in an (infinite) *2-dimensional grid* all nodes have degree 4, similar to a collection of cliques of size 5. The grid obviously forms a single connected component whereas the 5-cliques are not connected at all. Thus it is desirable to use additional measures for testing the similarity of two networks more accurately.

A more refined measure of structural similarity is achieved by comparing the ℓ -hop degree distribution of the general duplication model and the yeast proteome network. In a given network, the ℓ -hop degree of a node is defined to be the total number of unique nodes it can reach in at most ℓ hops. Clearly the 1-hop degree of a node is its own degree.

In Figure 8 we plot the average ℓ -hop degree of nodes as a function of their degree, both for the general duplication model and the yeast proteome network. By definition, the 1-hop degree distribution is a straight line with slope 1. Notice that for $\ell > 2$ the ℓ -hop degree distribution of the yeast proteome network is very different from that of the general duplication model. In fact, for $\ell > 2$, the number of nodes that can be reached by a typical node in the yeast proteome network is much higher than that observed in the general duplication model. We observed this qualitative difference for the general duplication model with all parameter choices we tested.

In order to capture the ℓ -hop degree distribution of the yeast proteome network for $\ell > 2$, we develop a more refined model that aims to emulate the divergence mechanisms in proteome network evolution more accurately. This model, which we call the *sequence similarity enhanced model*, exhibits a degree distribution very similar to that of the yeast network while also capturing its ℓ -hop

degree distribution as seen in Figure 8. We provide the details of our enhanced model in the next section.

3.1 Sequence Similarity Distribution in the Yeast Proteome

A mathematical model for capturing proteome network evolution should take into account Ohno’s theory, which attributes the genome sequence growth and evolution to subsequent gene duplications followed by mutations on the gene sequences. The general duplication model implements gene duplications through a uniformly random node selection process. The mutations are implemented through random edge deletions and insertions. A more refined mutation model may take into account the sequence composition of genes and their associated proteins, and also their pairwise similarity levels.

Given two protein sequences A and B , one way to measure their similarity level through the use of their global alignment score $S(A, B)$ based on the BLOSSOM62 scoring matrix - the default scoring matrix for the best used protein alignment tools. The normalized similarity score of A and B can then be defined as

$$SN(A, B) = \frac{S(A, B)}{S(A, A) + S(B, B) - S(A, B)}.$$

Clearly $SN(A, A) = 1 = 100\%$. Note that $1 - SN(A, B)$ forms a metric, which turns out to be quite useful for our purposes.

Once the similarity between two proteins is determined via the above measure, one can depict how protein sequences relate to each other by plotting the distribution of their pairwise similarities. Such plots are provided for the yeast proteome in Figures 4, 5. The yeast genome was downloaded from *Saccharomyces Genome Database* [3] and the pairwise alignment of the ≈ 6700 protein coding sequences were computed via *FASTA align* [27] with default parameters. In Figure 4 we display the number of protein pairs whose normalized similarity score is in the range $x\% + 0.05$ for varying values of x . One can observe that the pairwise similarity distribution has a peak value at $\sim 50\%$ followed by a very sharp drop.

The same distribution is depicted in a different perspective in Figure 5. Here the number of protein pairs whose normalized similarity score is *at least* $x\%$ is plotted for $x \in [20, 100]$. Observe that most pairs have a similarity score below a threshold value $\sim 50\%$ and comparatively very few pairs have a similarity score above that threshold value. The step function behavior of the normalized similarity score suggests that the pairs of proteins can be divided into two classes: protein pairs which are *similar* are the ones whose similarity scores are *above* the threshold; the other protein pairs are *dissimilar*.

Through an investigation of the yeast proteome network we observed that sequence-wise “similar” proteins have similar interaction patterns.

More specifically, we considered all pairs of proteins A, B for which there is another protein C that is sequence-wise similar to A and which interacts with C . Among these protein pairs, the frequency of those which interact is 21 times the frequency of protein pairs that interact among *all* protein pairs.

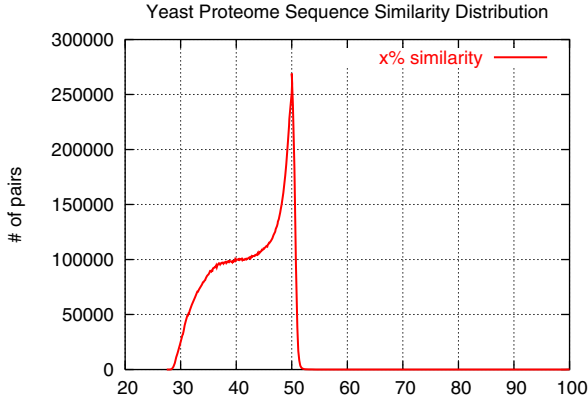


Fig. 4. Distribution of sequence similarity between pairs of yeast proteins (granularity: 0.1%)

Observation 1. *Given three proteins A, B, C , if A and B are sequence-wise similar and A interacts with C , then the chance that B interacts with C is ~ 21 times of that between arbitrary pair of proteins.*

Another observation we made was on the correlation between sequence similarities of protein triplets:

Observation 2. *Given three proteins A, B, C , if $A - B$ and $B - C$ are pairwise similar, then with $\sim 65\%$ chance $A - C$ are similar.*

This observation is not very surprising as the normalized similarity score above forms a metric and the number of protein pairs whose similarity score is above the threshold value is distributed uniformly over the range $[50\% - 100\%]$. Nevertheless it will be quite useful in establishing our enhanced proteome growth model which we describe in the next section.

3.2 Enhanced Model Based on Sequence Similarity

Based on our observations on the sequence similarity and its implications on protein-protein interactions we develop a more refined network generation model below. Our new model, which we call the *sequence similarity enhanced model*, modifies the step for updating the interaction edges of a duplicated node through the use of additional edges indicating sequence similarity. Thus the new model has two types of edges: *interaction edges* connecting proteins that interact with each other, and *sequence similarity edges* connecting proteins that are similar.

As per the general duplication model, our sequence similarity enhanced model works in discrete time steps. Let $G(t - 1)$ be the network at the end of time step $t - 1$. At each time step t , a new node v_t is generated, again by picking one of the nodes w in $G(t - 1)$ uniformly at random and “duplicating” it to create v_t ;

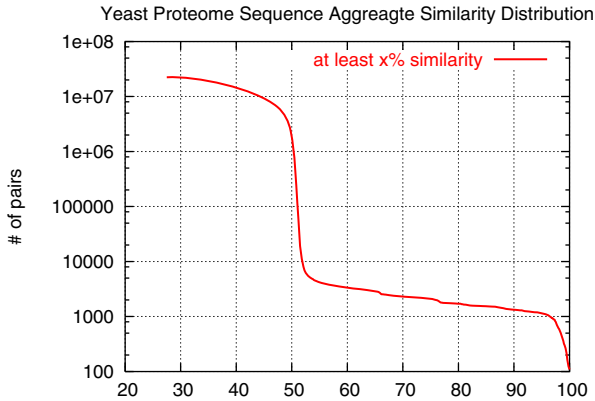


Fig. 5. Aggregate distribution of sequence similarity between yeast protein pairs. (aggregation performed from right to left).

i. e. v_t will initially be connected to all *similarity* neighbors and the *interaction* neighbors of v_t . The new node v_t will also be connected to w by a similarity edge. The following random process updates the similarity edges of v_t :

1. The similarity edge between v_t and w is deleted with probability δ .
2. Each remaining similarity edge is considered independently and is deleted with probability q' ($= 1 - p'$).
3. For each pair of similarity edges (v_t, u) and (u, u') , a similarity edge (v_t, u') is created with probability $(p')^2$.

Now the interaction edges of v_t are updated:

1. Each interaction edge is considered independently and is deleted with probability q ($= 1 - p$).
2. For each node u , which is not initially connected to v_t , a new edge (u, v_t) is created independently with probability r/t .
3. For each interaction edge (v_t, u) and each similarity edge (u, u') , a new interaction edge (v_t, u') is created with probability .03 (~ 21 times the chance of having an interaction edge between an arbitrary pair of nodes - following Observation [1](#)).

At the time of duplication, v_t and w are sequence-wise identical and thus each similarity edge (u, w) is duplicated as (u, v_t) . Step (i) of the similarity edge update process maintains the edge (w, v_t) with probability $1 - \delta$. Here, the parameter δ is the measure of divergence. In other words, the mutation events that occurred after the duplication event are reflected on the new edge by the deletion parameter *delta*. Step (ii) maintains every other similarity edge (u, v_t) with probability p' . Finally, Step (3) imposes Observation [2](#) on the constructed network. The interaction edge update process, in particular Steps (i) and (ii), works similar to that in the general duplication model. The only difference is in

Step (iii) where similarity edges are used to update interaction edges in order to impose Observation [1](#) on the constructed network.

The sequence similarity edges in the network are determined by two parameters, δ and p' . It is possible to estimate the values of δ and p' in the Yeast proteome network by fitting the *sequence similarity* degree distribution of the model with the *sequence similarity* degree distribution of the yeast proteome network. The best fitting sequence similarity degree distribution is achieved for $\delta = 0.7$ and $p' = 0.225$ and is given in Figure [6](#).

Based on the above values of δ and p' , it is possible to estimate the other two parameters, r and p that determine the interaction edges. The best fitting

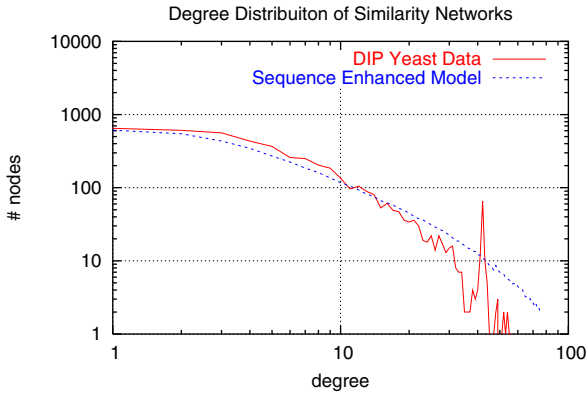


Fig. 6. The degree distribution of the proteome sequence similarity network of the yeast and that of the enhanced model with parameters $\delta = 0.7$ and $p' = 0.225$

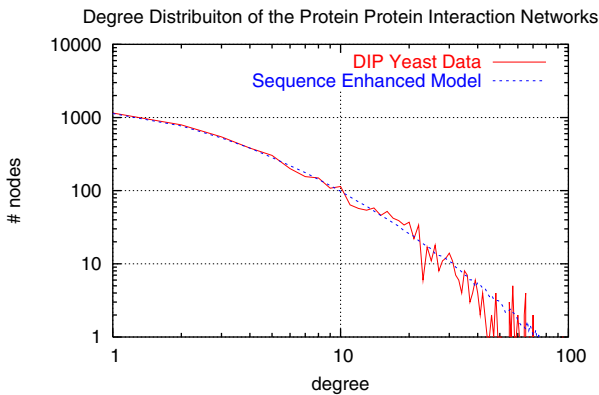


Fig. 7. The degree distribution of the proteome interaction network of the yeast and that of the enhanced model with parameters $q = 0.6$, $r = 0.1$, $\delta = 0.7$ and $p' = 0.225$

interaction degree distribution of the model to that of the yeast proteome network is achieved at $q = 0.6$ and $r = 0.04$ and is given in Figure 7.

The average clustering coefficients of the Yeast proteome network, the general model and the enhanced model can be seen in Table 1. The average clustering coefficients for the models are calculated using the resulting networks that have the best fitting degree distributions with the Yeast proteome network. In Figure 8, we finally compare the ℓ -hop degree distributions of the sequence similarity enhanced model, the general duplication model, and the yeast proteome network. Our sequence similarity enhanced model accurately captures the ℓ -hop degree distribution of the yeast proteome network for all values of ℓ .

Table 1. The average clustering coefficients of the DIP Data, and the models

	<i>Clustering Coefficient</i>
DIP Data	0.39
General Model	0.33 ± 0.01
Enhanced Model	0.37 ± 0.01

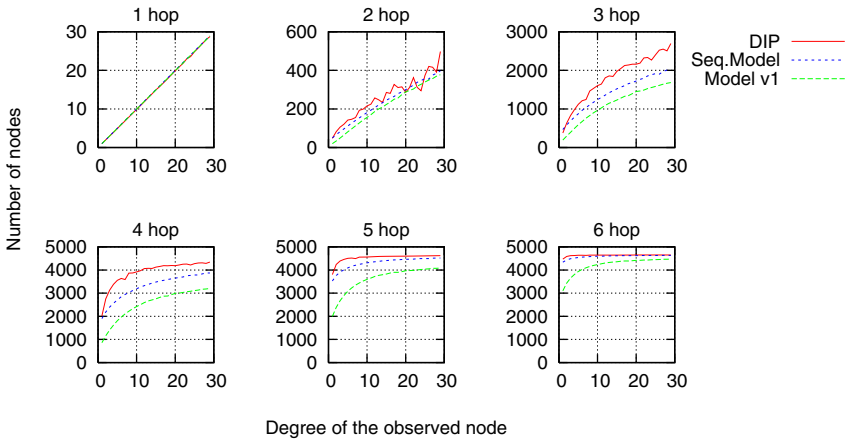


Fig. 8. The k -hop degree distribution of (i) the Yeast proteome network, (ii) the general duplication model, and (iii) the sequence similarity enhanced model. A typical node reaches all nodes in the network in 7 hops, thus ℓ -hop degree distribution for $\ell \geq 7$ is not very meaningful.

4 Conclusion

The paper first shows that the degree distribution of the pure duplication model ($r = 0$) cannot be a power law as stated in [10]. Then it shows that the degree distribution of the general model can not be a power law with exponential cut-off as stated in [26]. These two problems have been addressed in [5] where the

general duplication model for $r > 0$ is established to have a power law degree distribution. Unfortunately, in this paper, we observe that the general duplication model does not capture the more general ℓ -hop degree distribution of the yeast proteome network for $\ell > 1$. Thus, a new model, which takes into account the sequence similarity between protein pairs as a binary relationship, in addition to their interactions is introduced. This new model is shown to accurately capture the ℓ -hop degree distribution of the yeast interaction network for all $\ell > 0$ in addition to yielding a good approximation to the degree distribution of the yeast similarity network.

References

1. Aiello W., Chung F., Lu L., A random graph model for power law graphs, *Proc. ACM STOC*, pp 171-180, 2000.
2. Aiello W., Chung F., Lu L., Random evolution in massive graphs, *Proc. FOCS*, pp 510-519, 2001.
3. Balakrishnan, R., Christie, K. R., Costanzo, M. C., Dolinski, K., Dwight, S. S., Engel, S. R., Fisk, D. G., Hirschman, J. E., Hong, E. L., Nash, R., Oughtred, R., Skrzypek, M., Theesfeld, C. L., Binkley, G., Lane, C., Schroeder, M., Sethuraman, A., Dong, S., Weng, S., Miyasato, S., Andrada, R., Botstein, D., and Cherry, J. M. "Saccharomyces Genome Database" <ftp://ftp.yeastgenome.org/yeast/> (April 1, 2004).
4. Barabási, A.-L., Albert, R. A., Emergence of scaling in random networks, *Science* **286**, pp 509-512, 1999.
5. Bebek, G., Berenbrink, P., Cooper, C., Friedetzky, T., Nadeau, J.H., and Sahinalp, S.C., The degree distribution of the generalized duplication model. *Theoretical Computer Science*, 369:234-249, 2006.
6. Berger N., Bollobás, B., Borgs C., Chayes J., Riordan O., Degree distribution of the FKP network model, *Proc. ICALP*, LNCS 2719, pp 725-738, 2003.
7. Bhan A., Galas D. J., & Dewey T. G., A duplication growth model of gene expression networks, *Bioinformatics*, **18**, pp 1486-1493, 2002.
8. Bollobás, B., Borgs C., Chayes J., Riordan O., Directed scale-free graphs, *Proc. ACM-SIAM SODA*, pp 132-139, 2003.
9. Bollobás, B., Riordan, O., Spencer, J., and Tusanády, G., The degree sequence of a scale-free random graph process, *Random Structures and Algorithms*, **18**, pp 279-290, 2001.
10. Chung, F., Lu L., Dewey T.G., Galas D.J., Duplication models for biological networks, *Journal of Computational Biology*, **10**, pp 677-687, 2003.
11. Cooper C., Frieze A., A general model of webgraphs, *Random Structures and Algorithms*, **22(3)**: pp 311-335, 2003.
12. Deane, C.M., Salwinski, L., Xenarios, I. and Eisenberg, D., Protein interactions: Two methods for assessment of the reliability of high-throughput observations *Molecular and Cellular Proteomics* **1**:349-356, 2002.
13. Erdős, P., Rényi, A., On random graphs I, *Publicationes Mathematicae Debrecen*, **6**, pp 290-297, 1959.
14. Faloutsos M., Faloutsos P., Faloutsos C., On Power-Law Relationships of the Internet Topology, *SIGCOMM*, 1999.
15. Ferrer i Cancho, R., Janssen, C., The small world of human language, *Procs. Roy. Soc. London B*, **268**, pp 2261-2266, 2001.

16. Force A., Lynch M., Pickett F.B., Amores A., Yan Y., Postlethwait J., Preservation of duplicate genes by complementary degenerative mutations. *Genetics*, **151**, pp 1531-1545, 1999.
17. Han, J.D., Dupuy, D., Bertin, N., Cusick, M., and Vidal M., Effects of sampling on the predicted topology of interactome networks, *Nature Biotechnology* **23**, 839-844, (2005)
18. Ispolatov, I., Krapivsky, P.L., Yuryev, A., Duplication-divergence model of protein interaction network, *Physical Review*, E **71**, 061911, 2005.
19. Ito, T. et al., A Comprehensive two-hybrid analysis to explore the yeast protein interactome, *PNAS*, vol. **98**, no **8** pp 4569, 2001.
20. Jeong, H., Mason, S., Barabasi, A.-L. & Oltvai, Z. N., Lethality and centrality in protein networks, *Nature*, **411**, pp 41, 2001.
21. Kleinberg, J., Kumar, R., Raphavan, PP, Rajagopalan, S. and Tomkins, A., The Web as a graph: Measurements, models and methods, *Proc. COCOON*, Tokyo, Japan, pp 1-17, 1999.
22. Kumar R., Raghavan P., Rajagopalan S., Sivakumar D., Tomkins A., Upfal E., Stochastic models for the web graph, *Proc. FOCS* pp 57-65, 2000.
23. Nadeau, J.H., Sankoff D., Comparable Rates of Gene Loss and Functional Divergence After Genome Duplications Early in Vertebrate Evolution, *Genetics*, **147**, pp 1259, 1997.
24. van Noort, V., Snel, B., Huymen, M. A., The yeast coexpression network has a small-world scale-free architecture and can be explained by a simple model, *EMBO Reports*, Vol. **5**, No. 3, 2004.
25. Ohno, S., *Evolution by gene duplication*. Berlin: Springer, 1970.
26. Pastor-Satorras, R., Smith, E., and Sole, R.V., Evolving protein interaction networks through gene duplication, *J. Theor. Biol.* **222**, pp 199-210, 2003.
27. Pearson, W. R., Lipman, D. J. "Fasta" <ftp://ftp.virginia.edu/pub/fasta/> (data of access).
28. Przulj, N., Corneil, D. G., and Jurisica, I., Modeling Interactome: Scale-Free or Geometric?, *Bioinformatics*, vol.**20**, number 18, pages 3508-3515, 2004.
29. Redner, S., How Popular is Your Paper? An Empirical Study of the Citation Distribution, *Eur. Phys. Jour.* **B 4**, pp 131-134, 1998.
30. Seoighe C., Wolfe K.H., Yeast genome evolution in the post-genome era. *Current Opinion in Mol. Biol.*, **2**, pp 548-554, 1999.
31. Seoighe C., Wolfe K.H., Updated map of duplicated regions in the yeast genome. *Gene*, **238(1)**, 253-61, 1999.
32. Simon, H. A., On a class of skew distribution functions, *Biometrika*, **42**, pp 425-440, 1955.
33. Uetz, P. L. et al., A comprehensive analysis of protein-protein interactions in *Saccharomyces Cerevisiae*, *Nature*, **403**, pp 623-7, 2000.
34. Vázquez, A., Flammini, A., Maritan, A. & Vespignani, A., Modelling of protein interaction networks, *Complexus* **1**, 38-44, 2003.
35. Wagner, A., The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes, *Mol. Biol. Evol.* **18**, pp 1283-1292, 2001.
36. Watts, D. J. & Strogatz, S. H., Colective dynamics of small-world networks, *Nature*, **393**, pp 440-442, 1998.
37. Wolfe K.H., Shields D.C., Molecular evidence for an ancient duplication of the entire yeast genome, *Nature*, **387**, pp 708-713, 1997.
38. Xenarios, I. et al., DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions, *Nucleic Acids Res.* **30**, pp 303-305, 2002.

Application of Expectation Maximization Clustering to Transcription Factor Binding Positions for cDNA Microarray Analysis

Chih-Yu Chen^{1,3}, Von-Wun Soo^{1,2}, and Chi-Li Kuo¹

¹ Department of Computer Science, National Tsing Hua University, Taiwan

² Department of Computer Science and Information Engineering,
National University of Kaohsiung, Taiwan

³ Division of Biostatistics and Bioinformatics,
National Health Research Institutes, Taiwan
`jubilee@nhri.org.tw`

Abstract. We conduct the transcription factor (TF) analysis by detecting transcription factor pairs and incorporating binding positions for genes with altered expressions in time-series cDNA microarray data. Prediction of TF pairs that mostly likely contribute to the regulated transcription of differentially expressed genes are done through the computation of their expression coherence (EC). The Expectation Maximization (EM) clustering is performed additionally in order to detect patterns in specific TF binding positions. We evaluate the EC of expression profiles of genes within each cluster to discover binding trends that may play a significant role in regulation of target genes. Our method has successfully identified TF pairs that have a greater potential for regulating their target genes at specified locations rather than at arbitrary locations.

1 Introduction

The binding of transcription factors to specific binding sites (BS) in the promoter and enhancer regions contributes to differential expressions in target genes. Once the TFs are bound to their corresponding binding sites, activation domains of TFs interact with components such as RNA polymerase complex and accessory factors, and subsequently alter the expression of their target genes through direct gene-protein or indirect protein-protein interactions. Gene expression is controlled by combinatorial regulation of more than one transcription factor. Since BS's are only composed of around 8 to 20 nucleotides, the rest of the regions in TF hanging in space can interact with parts of other neighboring TFs. The interactions between transcription factors are significant in a sense that a TF bound to the promoter region may inhibit or trigger the engagement between another TF and the corresponding BS.

Composite regulatory elements [1] are two closely situated binding sites for different transcription factors located in transcription regulatory regions. Since most of them contain different protein domain structures, namely DNA-binding

and activation domains, composite elements bring about the combinatorial nature of transcription regulation. It is also notable that TFs with binding sites distantly situated might still be able to interact due to the looping of DNA by some other TFs, so binding sites of TFs do not always have to be closely situated for an interaction to occur [2].

There are works in search of TFs which bind cooperatively [3] [4] without gene expression analysis. The main aim of cDNA microarray technique is to find out the differences of transcription activities along with enormous genes from cells in experiments or biopsy. Conventional analyses on microarray data pick out genes showing differential expressions in the profile, biologists then categorize and link the functions of those genes to the experiment in order to understand the gene-protein interaction and cellular activities. There have already been researches investigating human diseases and therapeutics in relation to transcription factor activities [5] [6]. Therefore, incorporating the regulatory mechanism of transcription factor is an essential element for providing new insights in biological regulations. In this study, expression patterns of specific TF-targeted genes from time-series expression data set will be clustered for prediction of predominate dual transcription factors and their binding positions that most likely contribute to the expression changes.

Expression coherence (EC) is a measure of the overall similarity of the expression profiles of all the genes containing the specific putative BS of a given TF(s). Pilpel [7] conducted an exhaustive search on significantly synergistic motif pairs by calculating EC scores (Appendix I) from expression profiles of the yeast genes containing the TFBS pairs in the promoters (Fig. 1). His goal is to detect significant differences between the individual EC scores and the combinatorial EC score for each pair of TFs in several conditions. A pair of TFs are concluded to be synergistically regulating the gene transcription of a specific process if the expression profiles of target genes involved in the process are significantly coherent. This particular approach traces the upstream promoter regions of genes to build the connection between TFs and genes instead of using the expression profiles of TFs as an indicator of gene regulation [8]. The approach from Ref [8] also uses expression data to find TF complexes, but by disregarding non-differentially expressed TF coding genes, activities of TFs attributed mostly to post-transcriptional changes are ignored.

As an extension of the EC score approach, we incorporate the position ranges for TFBSs into the analysis, and instead of analyzing yeast data in different conditions, we apply the methods to human expression data under an external stimulation. Since TFs are known to physically interact with other factors, and sometimes induce or inhibit bindings of other factors to DNA, we suspect that the binding positions (BPs) of TFs may play a substantial role in combinatorial regulation. We conduct the position analysis by making the assumption that the binding position is a crucial determinant of TF complex formation, as previously noted in [1] [2] [6] [7]. Under the assumption, TFs binding at specific ranges can physically interact with one another, which in turn influence the transcription of the target genes. In short, we present an alternative way of analyzing time-series

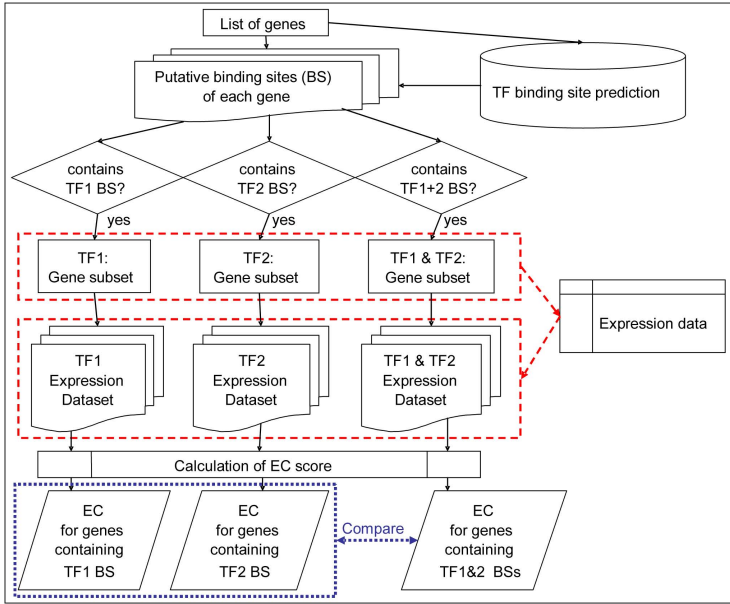


Fig. 1. Expression coherence comparison from Pilpel [7]

microarray data (Fig. 2), and the objective is to detect regularities in BPs of the dual TFs that are responsible for a consistent pattern in expression profiles of their co-target genes.

2 Methods

Our method is applied to microarray data retrieved from Stanford Microarray Database [9]: prostate cancer cell line with methylseleninic acid (MSA) treatments [10]. The time-series data contain 3 dosages of MSA, and the expression values are taken at several different time points for each dosage. We retrieved the promoter sequences, 1000 nucleotides upstream of the transcription start site, of all human genes from Entrez Genes [11]. The binding specificities of TFs to the DNA binding sites are modeled using position specific scoring matrices (PSSMs) and these matrices are collected in databases such as TRANSFAC [12] and JASPAR [13]. We downloaded 108 Homo sapiens PSSMs freely distributed from TRANSFAC database in March 2005. We then compute putative TFBS in all promoter sequences utilizing the TRANSFAC matrices and the same computation method from TFSearch (<http://www.cbrc.jp/research/db/TFSEARCH.html>). First we collect a list of genes with a logarithmic ratio greater than 1.5 at any time point, and then perform the analysis in Figure 2.

In order to catch the regularities in binding positions of two TFs, an EM algorithm [14] [15] is applied to discover clusters in positions of TF1 and TF2. We

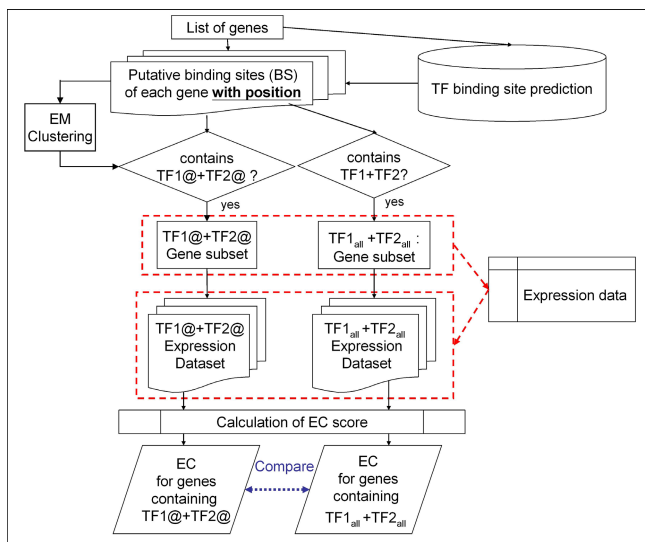


Fig. 2. Proposed Method for Position Analysis. TFBPs located within the specific range computed by EM clustering are denoted with “@” and TFBPs situated throughout the retrieved 1000 nucleotides (nt) are denoted as TF1_{all} and TF2_{all}.

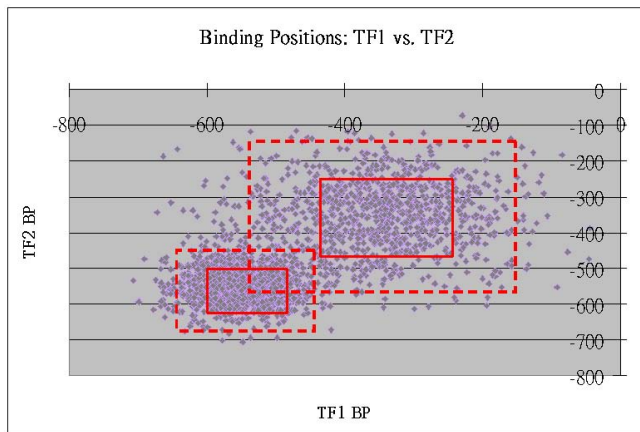


Fig. 3. BP scatter plot: Application of EM on binding positions using Weka [10]

assume the probability distribution of the binding positions of dual TFs to the promoter regions of a given list of genes is composed of Gaussian distributions. Figure 3 shows an arbitrary example of TF1 and TF2 binding positions in an arbitrary list of genes. TF1 BPs are represented in the x-axis in the scatter plot and the respective TF2 BPs are represented in the y-axis. Each dot represents the BPs of TF1 and TF2 in the promoter region of a gene. Since we retrieved

1000 upstream nts, the binding positions range from 0 to -1000, 0 being the transcription start site and -1000 being 1000 bases upstream. In our case, EM clustering helps detecting condensed regions where common locations of TF1 and TF2 BS's are frequent between genes.

EM algorithm helps us estimate the most proper parameter values in the Gaussian distributions and thus estimate the most likely binding positions of dual TFs. For example, the two clusters computed by EM are boxed in Figure 3 with the outer rectangles being the range computed from 2 standard deviations (sd) and the inner rectangles indicating the range computed from 1 sd. The expression profiles of genes found within each cluster are then extracted from the microarray data to calculate the clustered EC scores (Fig. 2). In order to avoid overlaps, the range of positions used for the EM cluster is (Mean-1sd, Mean+1sd), which is the range of the inner rectangle. Finally the clustered scores are compared to the overall scores for TF1 and TF2 combined, and clusters with better expression coherence are identified. Details on the computational method of EM clustering are provided in Appendix II.

3 Results

Figure 4 is an example output from the method in Figure 2. It shows that specifying Sp1 and USF BPs within the specific ranges can improve the expression

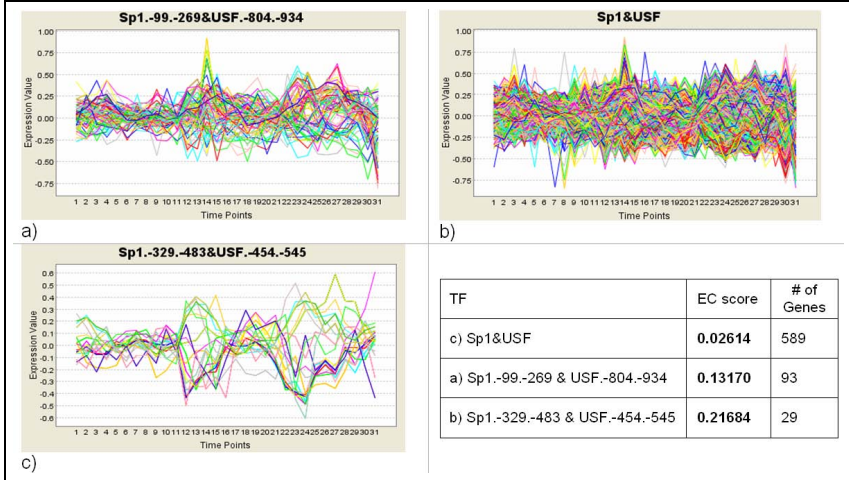


Fig. 4. The position analysis on Sp1 & USF

Table 1. The corresponding time points

Dosage	3μM MSA								0μM	10μM MSA								30μM MSA													
Time(hr)	1	2	4	6	9	12	15	18	24	48	0	1	2	4	6	9	12	15	18	24	48	1	2	4	6	9	12	15	18	24	48
Label	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

coherence of the target genes. Each line in Figure 4a), for example, represents the expression profile of a gene targeted by both Sp1 and USF within the position range of (-99, -269) and (-804, -934) in the promoter region, respectively. The y-axis represents normalized expression values and the corresponding time points on the x-axis are listed in Table 1.

In Figure 4c), the expression coherence of Sp1 and USF in general has a low EC score of 0.02614, so the relation between the two TFs can be overlooked in absence of our proposed binding position analysis. The cluster that specifies Sp1 BP within (-99, -269) and USF BP within (-804, -934) obtained an EC score of 0.13170, which is around 5 times the EC score of the dual TFs in general. The other cluster of genes in Figure 4c) obtained a score of 0.21684, which is more than 8 times the general score. This indicates that genes regulated by these two TFs within the position ranges have better coherence in expression than genes regulated by the two TFs at arbitrary positions. The scores improve dramatically in the two position clusters outputted from EM, which means that these 2 TFs binding at specific ranges of positions stated in Figure 4 have a better chance of contributing to regulations of co-target genes. The interaction between Sp1 and USF has been experimentally proven to exist in tumor cells [16], therefore, further potential interactions between 2 TFs can be detected in our proposed position analysis.

We have also found, as shown in Fig. 5, a potential interaction between Sp1 BP within (-58,-200) and NF-kappa B (NF- κ B) BP within (-41,-133). Again, the expression coherence of Sp1 and NF- κ B in general only has an EC score of 0.02829, so the relation cannot be identified using the original EC method [7]. NF- κ B and Sp1 has been confirmed by experimental studies in vivo to physically interact to each other, and work cooperatively in DNA binding and in transcriptional activation [17] [18]. NF- κ B BS between bases -43 and -77 and Sp1 BS between -129 and -180 of the M-CSF promoter were found to be important for basal transcription [19]. It has also been reported that Sp1 BS located 42 base pairs upstream from the NF- κ B BS (-65,-83) are essential for basal

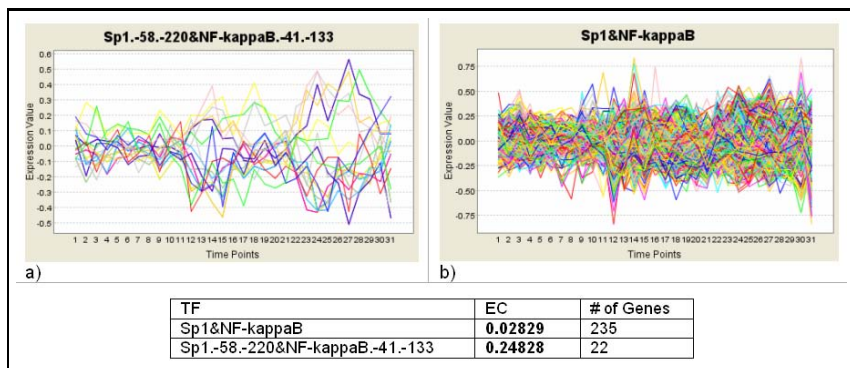


Fig. 5. The position analysis on Sp1 & NF-kappa B

MGSA/GROalpha promoter activity in melanoma [20]. The two TFs were also identified to regulate the promoter of $I\kappa B\alpha$, whose gene product retains NF- κ Bs in the cytoplasm until stimulation occurs [21]. Recently, they are found to activate p21 and FasL gene for leukemia cell survival [22] [23]. Table 2 lists TF pairs regulating more than 20 genes at the specified binding positions with the highest EC scores for the MSA data. We have demonstrated the second and the fifth places in the figures above with literature support.

Table 2. EM Position Analysis: Top 20 entries with greater than 20 target genes

TF1	Binding Pos.	TF2	Binding Pos.	EC	Gene Count
TATA	(-42,-182)	HFH-3	(-648,-913)	0.276667	21
Sp1	(-58,-220)	NF-kappaB	(-41,-133)	0.248276	22
AP-1	(-93,-293)	TATA	(-730,-918)	0.247863	25
SRY	(-246,-747)	HLF	(-23,-111)	0.229437	20
Sp1	(-329,-483)	USF	(-454,-545)	0.216837	29
NF-Y	(-108,-274)	AREB6	(-899,-964)	0.206061	28
SRY	(-250,-758)	NF-Y	(-937,-980)	0.200784	25
AP-4	(-196,-518)	YY1	(-234,-659)	0.185185	23
C/EBPbeta	(-38,-140)	C/EBP	(-370,-787)	0.175403	23
p53	(-215,-795)	c-Myc/Max	(-172,-731)	0.172308	24
Sp1	(-55,-307)	SREBP-1	(-678,-914)	0.170455	34
Sp1	(-102,-419)	YY1	(-864,-967)	0.168067	21
Sp1	(-171,-594)	Egr-3	(-50,-137)	0.166154	30
SRY	(-153,-514)	SREBP-1	(-701,-891)	0.165531	48
GATA-2	(-632,-892)	SREBP-1	(-65,-286)	0.162879	30
GATA-2	(-49,-190)	FOXJ2	(-259,-631)	0.162562	29
CRE-BP1/c-Jun	(-476,-892)	Freac-7	(-343,-876)	0.162055	22
GATA-3	(-776,-933)	Tst-1	(-278,-800)	0.16129	26
NF-kappaB	(-197,-568)	MZF1	(-877,-945)	0.158621	21
TCF11	(-57,-285)	TGIF	(-657,-871)	0.156923	20

4 Discussion

Our proposed methods extend the TF modularity analysis by combining the EC score analysis with EM position clustering. As a result, our method efficaciously detected the relations between Sp1 and USF, as well as Sp1 and NF- κ B that could not be identified using the method from Ref [7] alone. As a result, we have made a connection between the drug MSA, cancerous cells and the regulating dual TFs as well as the coherently expressed target genes. Additionally, we provide an supplementary way to discover potential interactions between dual TFs at specific positions that could account for the regulation of target genes.

However, TF modules may comprise of several TFs, and our method can only identify combinations of 2 TFs due to computational limitations. The limited information on transcription factors also restricted the binding site predictions to only those TFs with known matrices and even from that, binding sites can

still be overlooked by matrices with insufficient amount of experimental data in the TRANSFAC database. The neglected TF binding sites may play key roles in gene regulation and the corresponding TF may interact with other main TF components to induce or reduce transcription. These limitations are reflected in low EC scores, since dual TFs are typically not the only players in regulation.

With our binding position analysis, biologists can conduct further experiments on the genes that are targeted by the identified dual TFs at specific binding positions to validate the results and discover new relations. For the 108 known human TF matrices, there are 5778 pairs of TFs that could contribute to the global expression changes, and now that we can narrow down the search, we are one step closer to decipher the complex underlying mechanism of gene regulation. Furthermore, the significant position ranges computed could also be used to check for SNPs that may alter the binding affinity of the TF, which in turn, changes the target expression profiles.

Acknowledgement

This project is supported in part by the National genomic medicine project of National Science Council of Taiwan R.O.C. under the grant number NSC-92-3112-B-007-003.

References

1. O. V. Kel-Margoulis, A. E. Kel, I. Reuter, I. V. Deineko, and E. Wingender, "TRANSCmpel: a database on composite regulatory elements in eukaryotic genes," *Nucleic Acids Res*, vol. 30, pp. 332-4, 2002.
2. K. Ogata, K. Sato, and T. H. Tahirov, "Eukaryotic transcriptional regulatory complexes: cooperativity from near and afar," *Curr Opin Struct Biol*, vol. 13, pp. 40-8, 2003.
3. D. GuhaThakurta and G. D. Stormo, "Identifying target sites for cooperatively binding factors," *Bioinformatics*, vol. 17, pp. 608-21, 2001.
4. R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp, "CREME: a framework for identifying cis-regulatory modules in human-mouse conserved segments," *Bioinformatics*, vol. 19 Suppl 1, pp. i283-91, 2003.
5. J. G. Emery, E. H. Ohlstein, and M. Jaye, "Therapeutic modulation of transcription factor activity," *Trends Pharmacol Sci*, vol. 22, pp. 233-40, 2001.
6. J. Villard, "Transcription regulation and human diseases," *Swiss Med Wkly*, vol. 134, pp. 571-9, 2004.
7. Y. Pilpel, P. Sudarsanam, and G. M. Church, "Identifying regulatory networks by combinatorial analysis of promoter elements," *Nat Genet*, vol. 29, pp. 153-9, 2001.
8. E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nat Genet*, vol. 34, pp. 166-76, 2003.
9. C. A. Ball, I. A. Awad, J. Demeter, J. Gollub, J. M. Hebert, T. Hernandez-Boussard, H. Jin, J. C. Matese, M. Nitzberg, F. Wymore, Z. K. Zachariah, P. O. Brown, and G. Sherlock, "The Stanford Microarray Database accommodates additional microarray platforms and data formats," *Nucleic Acids Res*, vol. 33, pp. D580-2, 2005.

10. H. Zhao, M. L. Whitfield, T. Xu, D. Botstein, and J. D. Brooks, "Diverse effects of methylseleninic acid on the transcriptional program of human prostate cancer cells," *Mol Biol Cell*, vol. 15, pp. 506-19, 2004.
11. D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez Gene: gene-centered information at NCBI," *Nucleic Acids Res*, vol. 33, pp. D54-8, 2005.
12. E. Wingender, X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhauser, M. Pruss, F. Schacherer, S. Thiele, and S. Urbach, "The TRANSFAC system on gene expression regulation," *Nucleic Acids Res*, vol. 29, pp. 281-3, 2001.
13. D. Vlieghe, A. Sandelin, P. J. De Bleser, K. Vleminckx, W. W. Wasserman, F. van Roy, and B. Lenhard, "A new generation of JASPAR, the open-access repository for transcription factor binding site profiles," *Nucleic Acids Res*, vol. 34, pp. D95-7, 2006.
14. I. H. Witten, and Frank, E., *Data Mining: Practical Machine Learning Tools with Java Implementations*. San Francisco: Morgan Kaufmann, 2000.
15. P. Bradley, Fayyad, U., and Reina, C., "Scaling EM (Expectation Maximization) Clustering to Large Databases," Microsoft Research 1998.
16. Y. Ge, T. L. Jensen, L. H. Matherly, and J. W. Taub, "Physical and functional interactions between USF and Sp1 proteins regulate human deoxycytidine kinase promoter activity," *J Biol Chem*, vol. 278, pp. 49901-10, 2003.
17. N. D. Perkins, N. L. Edwards, C. S. Duckett, A. B. Agranoff, R. M. Schmid, and G. J. Nabel, "A cooperative interaction between NF-kappa B and Sp1 is required for HIV-1 enhancer activation," *Embo J*, vol. 12, pp. 3551-8, 1993.
18. N. D. Perkins, A. B. Agranoff, E. Pascal, and G. J. Nabel, "An interaction between the DNA-binding domains of RelA(p65) and Sp1 mediates human immunodeficiency virus gene activation," *Mol Cell Biol*, vol. 14, pp. 6570-83, 1994.
19. R. A. Sater, "Basal expression of the human macrophage colony-stimulating factor (M-CSF) gene in K562 cells," *Leuk Res*, vol. 18, pp. 133-43, 1994.
20. L. D. Wood, A. A. Farmer, and A. Richmond, "HMGI(Y) and Sp1 in addition to NF-kappa B regulate transcription of the MGSA/GRO alpha gene," *Nucleic Acids Res*, vol. 23, pp. 4210-9, 1995.
21. M. Algarte, H. Kwon, P. Genin, and J. Hiscott, "Identification by in vivo genomic footprinting of a transcriptional switch containing NF-kappaB and Sp1 that regulates the IkappaBalpha promoter," *Mol Cell Biol*, vol. 19, pp. 6140-53, 1999.
22. J. Savickiene, G. Treigyte, A. Pivoriunas, R. Navakauskiene, and K. E. Magnusson, "Sp1 and NF-kappaB transcription factor activity in the regulation of the p21 and FasL promoters during promyelocytic leukemia cell monocytic differentiation and its associated apoptosis," *Ann N Y Acad Sci*, vol. 1030, pp. 569-77, 2004.
23. J. Savickiene, G. Treigyte, K. E. Magnusson, and R. Navakauskiene, "p21 (Waf1/Cip1) and FasL gene activation via Sp1 and NFkappaB is required for leukemia cell survival but not for cell death induced by diverse stimuli," *Int J Biochem Cell Biol*, vol. 37, pp. 784-96, 2005.

A Appendix I

Expression coherence (EC) values of genes regulated by subsets of TFs are calculated to quantitatively evaluate the degree of expression unity. We adapt the method from Pilpel [7] which computes pair-wise Euclidean distances of the expression profiles of TF-targeted genes and calculates the fraction that is lower

than a threshold distance. The threshold is previously determined by calculating the Euclidean distances of a random sample of 100 genes and is set to the boundary value in the fifth percentile of the distribution of the distances. Expression profiles are normalized by multiplying all values of a gene by a scaling factor, so that the sum of the squares of all the values of each gene is 1.

```

For gene x, with index from 0 to  $N_i-2$  in  $S_i$ 
  For all genes, y, with index greater than x+1
    Compute  $d_{xy}$ .
    If  $d_{xy} \leq \text{Threshold}$ , count++.
  End of for
End of for

EC =  $\frac{\text{Count}}{N_i * (N_i - 1) / 2}$ 

```

Fig. 6. Computing EC scores for genes targeted by TF_i . For genes targeted by both TF_i and TF_j (dual TFs), simply replace S_i by S_{ij} and N_i by N_{ij} .

Let S_i be the set of genes targeted by TF_i , S_{ij} be the set of genes targeted by both TF_i and TF_j , and N_i/N_j be the number of genes in S_i/S_{ij} . The expression profiles of gene x and $y \in S_i$ are denoted by vectors E_x and E_y .

$$d_{xy} = |E_x - E_y| = \sqrt{\sum_{m=1}^n |E_{xm} - E_{ym}|^2}$$

where E_{xm} and E_{ym} are the m^{th} expression value of gene x and gene y respectively and n is the total number of experiments, $n = 31$ for the MSA data. Since d_{xy} is the square root of a positive value, d_{xy} is always greater than or equal to 0. The greater the value d_{xy} , the smaller the degree of coherence between E_x and E_y . Hence gene pairs with d_{xy} less than the threshold are considered close in distance and coherent in expression.

The total number of pairs in a set of N_i genes are $N_i * (N_i - 1) / 2$, so the number of Euclidean distances d_{xy} lower than the threshold is counted and divided by $N_i * (N_i - 1) / 2$. The scores evaluate how well TFs explain expression changes of the corresponding regulated genes. We compare the EC scores of each individual TF (TF_i or TF_j) with that of dual TFs (TF_i and TF_j) to see which pairs of TFs result in better EC scores. Higher EC scores, in definition, demonstrate higher chances of synergistic gene regulation by the dual TFs.

B Appendix II

The Expectation-Maximization (EM) [14] [15] is a mixture-based algorithm that maximizes the likelihood of the model. A mixture is a set of n probability distributions, in this case Gaussian distributions, where each of them is a component

with a density function f_d , $d = 1, \dots, n$. EM models the density distribution of instances in G containing m records and each instance belongs to a component with its density function. In our case, each component represents a cluster in the binding position scatter plot (Fig. 3) where each cluster is an indication of a frequent binding pattern of the dual TFs on their target genes.

Let there be t continuous attributes in G . In the case with two TF position clustering, t is equal to 2: the binding positions (BPs) of TF₁ and TF₂. With the assumption that the attributes are independent random variables, the probability functions for all attributes are multiplied together to obtain the joint probability for the instance. Hence the probability function for 2 attributes, TF₁BP and TF₂BP, is

$$f_d(x \mid \mu_{d_1}, \sigma_{d_1}, \mu_{d_2}, \sigma_{d_2}) = f_{d_1}(x_1 \mid \mu_{d_1}, \sigma_{d_1}) * f_{d_2}(x_2 \mid \mu_{d_2}, \sigma_{d_2})$$

where $f_{d_1}(x_1 \mid \mu_{d_1}, \sigma_{d_1})$ is the density function of the Gaussian distribution for TF₁ and μ_{d_1}, σ_{d_1} represent the mean and the standard deviation for TF₁. For the ease of notation, we let $f_d(x \mid \mu_d, \sigma_d)$ denote the joint probability function for gene record x .

Let x be a gene record in G , the mixture model probability density function for x is

$$p(x) = \sum_{d=1}^n w_d \cdot f_d(x \mid \mu_d, \sigma_d)$$

where $w_d \geq 0$, the weight, represents the fraction of gene records in component d and $\sum_{d=1}^n w_d = 1$.

Components are allowed to overlap such that certain records may belong to multiple clusters with different weights. The individual weight of x in component d is given by

$$w_d(x) = \frac{w_d \cdot f_d(x \mid \mu_d, \sigma_d)}{\sum_{j=0}^n w_j \cdot f_j(x \mid \mu_j, \sigma_j)}$$

The algorithm is similar to the K-means method such that a set of parameters are recomputed until a desired convergence is reached. Let Θ denotes all the parameters for the mixture model. The overall likelihood of the data is a measure of cluster quality, which indicates how well the mixture model fits the data.

The overall likelihood is obtained by multiplying the probabilities of each gene record x :

$$\prod_{x \in G} p(x) = \prod_{x \in G} \left(\sum_{d=1}^n w_d \cdot f_d(x \mid \mu_d, \sigma_d) \right)$$

The likelihood has been theoretically proven to increase every iteration of the EM algorithm and greater likelihoods represent higher quality in clustering. The log-likelihood expression for the mixture model is

$$L(\Theta) = \log \prod_{x \in G} p(x) = \sum_{x \in G} \log[p(x)]$$

$$= \sum_{x \in G} \log \left(\sum_{d=1}^n w_d \cdot f_d(x \mid \mu_d, \sigma_d) \right)$$

The initial values for parameters Θ are estimated, and the algorithm updates them iteratively.

Step 1:

Given the initial guesses for Θ and a termination threshold ω , we use the probability density function for normal distributions to compute the weight (cluster probability) of instance x in each cluster $d = 1, \dots, n, \forall x \in G$. At iteration i ,

$$w_d^i(x) = \frac{w_d^i \cdot f_d(x \mid \mu_d^i, \sigma_d^i)}{\sum_{j=0}^n w_j^i \cdot f_j(x \mid \mu_j^i, \sigma_j^i)}$$

Step 2:

We use these probabilities to re-estimate the parameters Θ^{i+1} :

$$w_d^{i+1} = \sum_{x \in G} w_d^i(x), \quad \mu_d^{i+1} = \frac{\sum_{x \in G} (w_d^i(x) \cdot x)}{\sum_{x \in G} w_d^i(x)}$$

$$\sigma_d^{i+1} = \sqrt{\frac{\sum_{x \in G} w_d^i(x) (x - \mu_d^{i+1})^2}{\sum_{x \in G} w_d^i(x)}}$$

$i++$ and repeat Step 1.

Termination Condition: The iteration discontinues if the increase in log-likelihood becomes negligible.

$$L(\Theta^{i+1}) - L(\Theta^i) \leq \omega.$$

In short, we are interested in the model that maximizes the likelihood of the distributions given the data, so the EM algorithm is applied to detect clusters of positions of putative BS's from a list of genes.

Combinatorial Genetic Regulatory Network Analysis Tools for High Throughput Transcriptomic Data

Elissa J. Chesler¹ and Michael A. Langston²

¹ Life Sciences Division, Oak Ridge National Laboratory,
P.O. Box 2008, Oak Ridge, TN 37831-6124, USA

² Department of Computer Science, University of Tennessee,
Knoxville, TN 37996–3450, USA

Abstract. A series of genome-scale algorithms and high-performance implementations is described and shown to be useful in the genetic analysis of gene transcription. With them it is possible to address common questions such as: “are the sets of genes co-expressed under one type of conditions the same as those sets co-expressed under another?” A new noise-adaptive graph algorithm, dubbed “paraclique,” is introduced and analyzed for use in biological hypotheses testing. A notion of vertex coverage is also devised, based on vertex-disjoint paths within correlation graphs, and used to determine the identity, proportion and number of transcripts connected to individual phenotypes and quantitative trait loci (QTL) regulatory models. A major goal is to identify which, among a set of candidate genes, are the most likely regulators of trait variation. These methods are applied in an effort to identify multiple-QTL regulatory models for large groups of genetically co-expressed genes, and to extrapolate the consequences of this genetic variation on phenotypes observed across levels of biological scale through the evaluation of vertex coverage. This approach is furthermore applied to definitions of homology-based gene sets, and the incorporation of categorical data such as known gene pathways. In all these tasks discrete mathematics and combinatorial algorithms form organizing principles upon which methods and implementations are based.

Keywords: Microarray Analysis, Putative Co-Regulation, Quantitative Trait Loci, Regulatory Models.

1 Introduction

We describe ongoing research efforts aimed at developing, implementing and validating graph theoretical approaches and high-performance computing implementations for the systems genetic analysis of high throughput molecular phenotypes in relation to higher order systems-level traits. Present approaches to these problems typically deal only with individual genes or small sets of genes and a small handful of systems phenotypes. Early analytic approaches relied primarily on simplifying assumptions that only a single locus is involved in gene regulation [12][18][36]. This is despite widespread acknowledgment that gene expression is a complex phenotype regulated by multiple genetic and environmental factors. It has been simply a limitation of the commonly employed analytic tools, which only evaluate one locus at a time. Recent studies have systematically examined

two-locus interactions [11], but we have observed regulation by larger combinations of loci. To search the model space for the best multi-locus modeling, arbitrary filtering of the data is often required to reduce the problem size to a manageable size. Instead, we represent the entire data set in a graph theoretical context, and employ a unique top-down approach. We transform the expression genetic covariance matrix into a simple, undirected, unweighted graph, and extract pure groups of highly inter-correlated expression traits. These groups represent a much smaller set of traits that are then subject to mapping analysis. We examine the connectivity among these sets and analyze the molecular, biochemical and genetic regulatory commonality of connected genes using novel and existing bioinformatics tools. We also develop data-driven hypotheses to explain the mechanisms of genetic perturbations and variation as a means of defining global consequences of individual differences on tissue structure and function.

Much of our work is motivated by prior studies of brain gene expression and mRNA abundance levels. These are complex phenotypes regulated by multiple genetic and environmental factors [21]. We have previously performed genome-wide mapping¹ of the loci that modulate brain gene expression assayed on microarrays in a genetic reference population, and explored correlations of expression and complex phenotypes [20]. These and other transcriptome mapping studies have revealed several major “trans-bands,” that is, loci that regulate large numbers of transcripts encoded across the genome [33]. For example, we have recently identified at least seven loci that act in combination to regulate large numbers of transcripts encoding synaptic proteins and transcription or translation machinery [18]. Previous studies of genetic analysis of gene expression report trans-bands that reflect a high degree of covariance in the gene expression data [12,16,30,36].

2 Quantitative Trait Loci and Regulatory Models

The sources of genetic covariation, used here to define edge weights, are the genetic polymorphisms that occur naturally among individuals. These differences, acting first on the molecular scale, exert their effects through time, space and tissue compartment to influence traits as diverse as morphology, physiology and behavior. Because the genetic variation and covariation that we seek to explain is continuous (or quantitative) in nature, they are referred to as quantitative traits, determined by multiple genes and environmental conditions. To identify the genes (here literally referring to the heritable source of variation, not strictly to a particular class of genome features), experimental crosses are performed to shuffle genotypes from two genomes through meiotic recombination. For example, mouse strains C57BL/6J (B) and DBA/2J (D) are crossed, generating an F1 population with one copy of the B allele and one copy of the D allele at every location throughout the genome. These mice are crossed again to create an F2 generation, each member of which has a unique complement of shuffled B and D genomes, that is, they possess either two B alleles, a B and a D allele, or two D alleles at each locus. Quantitative techniques have been developed to associate the vectors

¹ Our transcriptomic data is publicly accessible in the WebQTL system www.webqtl.org. This tool allows systems genetic analysis of single genes or small sets of genes using a bottom-up approach.

of polymorphic states (genotypes) at known locations throughout the genome with the vector of phenotypes [23,34]. Statistically significant predictive genotype-phenotype relations define quantitative trait loci (QTLs). Because the marker is not typically the actual site of the polymorphism, interpolative methods have been developed to estimate the distance of the QTL from the marker and the strength of the association. Using multiple-regression and model-fitting methods, the true complexity of the phenotypic variation can be modeled through the consideration of multiple loci and environmental factors as predictors [13].

The typical experimental mapping population is bred once for each experiment, and the unique assortment of genotypes that result can never be retrieved. Mouse panels have been in use since the early 1980's, however, that can be used as a retrievable reference population. These populations, called recombinant inbred lines (henceforth RIL) consist of the progeny of an F2 cross that have been inbred for over twenty generations. The importance of these panels is that they allow population genetics methods to be applied to systems biology, by exploiting naturally occurring polymorphisms to determine the membership of biological networks that transcend time, space and tissue compartment. The largest of these sets contains eighty lines [35], and a large 1024 strain set is being bred at the Oak Ridge National Laboratory [22]. The genotypes are obtained at a very high precision and stored in public repositories such as www.genenetwork.org (encompassing WebQTL [19]) for analysis. Recently, genotypes have been identified at 15,000 loci using the Illumina SNP genotyping system (<http://www.well.ox.ac.uk/mouse/INBREDS/RIL/index.shtml>). Because the lines are inbred, attributes of the lines, whether they be genotypes, phenotypes, or high precision molecular trait data including microarray, can be aggregated indefinitely to form a single data matrix and analyzed using correlation techniques [20]. Phenotypic correlations can be partitioned into environmental and genetic sources of covariance. When the correlations are obtained on the phenotypic means based on subsampling of individuals within line, they can often be interpreted to be genetic correlations. This is particularly true if the trait data are obtained in independent sets of individuals. In transcription profiling, the multiple measures are observed in the same individuals, and it is possible, especially with low sample sizes, that these correlations are also largely environmentally driven.

The application of QTL mapping to microarrays, often termed "genetical genomics," was first reported in yeast [12], and has since been successfully performed in F2 [36] and RIL [16,18] mouse populations. In recent work [18], we have detected the presence of trans-QTL bands (locations in the genome that regulate hundreds of distally located transcript abundances) that regulate co-expression in the central nervous system. The trans-QTL bands are typically found by fitting single-locus models across the genome, and then, at locations throughout the genome, counting the number of transcripts for which the best fitting peak is found at that location. This bottom-up approach requires several assumptions that do not always hold, notably, that each transcript abundance is regulated by a single genetic locus. We have taken a top-down approach to this problem. We use graph algorithms first to decompose the genetic correlation matrix so as to identify sets of putatively co-expressed phenotypes, and then to determine the best multiple locus models to determine their regulation. Each of the sets of phenotypes that we

have identified is regulated by a combination of the trans-QTL bands. This top-down approach yields tremendous advantages due to the enormous computational demands that would be incurred if searching the entire model space.

Fundamental to this approach is the use of what we call “paracliques.” Informally, a paraclique is an extremely densely-connected subgraph, but one that may be missing a small number of edges and thus is not, strictly speaking, a clique. In the present application, this corresponds to a very highly intercorrelated group of genetically co-regulated genes whose transcript expression levels, as reflected in real and surely somewhat dirty microarray data, show highly significant but not necessarily perfect pair-wise correlations. By harnessing the computational power of tools such as fixed-parameter tractability, and then isolating paracliques, we are able to identify considerably denser subgraphs than are typically produced with traditional clustering algorithms. We have therefore reduced the immense genetic correlation matrix to a select set of intercorrelated modules, and have greatly simplified the discovery of functional significance and identity of genetic polymorphisms that underlie gene expression variation.

The RILs have been screened on over 1500 diverse phenotypes, and at least ten gene expression microarray profiling studies are completed or in progress in the RILs, which include several mouse, rat and plant species. The coverage of each paraclique (to be introduced in the sequel) by genotypes is determined. In this case, p -values are used as edge weights, due to the diverse sample sizes and correlation metrics applied to the data. The result is a graph of gene-to-phenotype relations. It consists of phenotype to expression relations, co-expression to regulatory models, and models to independent loci. See Figure 1.

3 Clique, Putative Co-regulation and Fixed-Parameter Tractability

We adopt graph theoretical approaches to analyze the huge correlation matrix that results from a microarray experiment that records expression levels over thousands of probes conducted over many conditions. The matrix is transformed into a complete graph, in which each gene is represented by a vertex, and in which each edge is weighted by the correlation coefficient of its endpoints. A suitable threshold, t , is then chosen. We are currently employing a variety of techniques to make this selection [32]. These include the use of functional similarity, ontological enrichment, known gene product interactions, and even methods based on spectral graph theory. A high-pass filter is applied to eliminate any edge whose weight is less than t . At this point, the weight of any remaining edge is ignored. This procedure produces a simple, undirected, unweighted graph for subsequent study. Note that all genes remain in the analysis. It is only the weakest correlations that are discarded.

A clique [10] in this new graph denotes a set of genes with the interesting property that every pair of its elements is highly correlated. This is widely interpreted as suggestive of putative co-regulation over the conditions in which the experiment was performed. Clique finding can be viewed as an especially stringent graph-theoretical form of clustering for gene co-expression data. Although clique presents an exceedingly difficult computational problem, its advantages are many. It is particularly note-

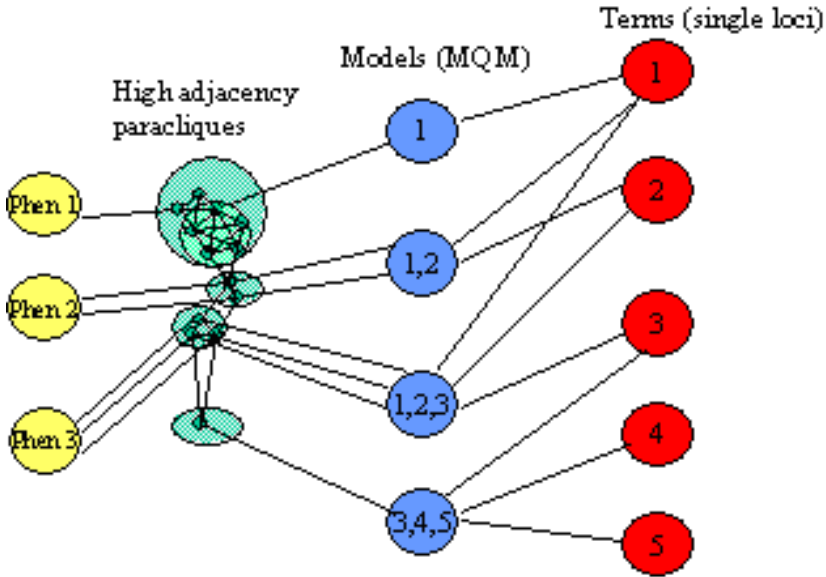


Fig. 1. Systems genetics can be viewed as a graph theoretical problem. With gene expression analysis (green), groups of co-expressed genes aka cliques are assembled into larger, disjoint paracliques. These can then be examined for coverage to phenotypes (yellow) and QTL models (blue) made of multiple single regulatory loci (red).

worthy that a vertex can reside in more than one clique, just as a gene can participate in more than one regulatory network. A huge variety of other clustering strategies are known that attempt to organize multivariate data into groups with approximately similar expression patterns [7,8,9,27,28,29,37]. Like ours, most methods build upon a correlation measure between expression levels to calculate a distance metric of similarity (or dissimilarity) of expression between each gene pair. There are several important limitations, however, to the vast majority of clustering algorithms that lie in contrast to the realities of biology. One such limitation is that the clusters these methods produce are disjoint, requiring that a gene be assigned to only one cluster. While this simplifies the amount of data to be evaluated, it places an artificial limitation on the biology under study because many genes play important roles in multiple but distinct pathways [14]. There are recent clustering techniques, for example those employing factor analysis [3], that do not require exclusive cluster membership for single genes. Unfortunately, these tend to produce biologically uninterpretable factors without the incorporation of prior biological information [26]. Another important limitation is that most of the measures of similarity used by current clustering algorithms do not permit the recognition of negative correlations, which are common and often equally meaningful from a biological perspective.

Ours is in fact a very general approach. Correlations between mRNA abundances and other molecular phenotype levels can be used as well to describe an edge-weighted graph, in which each vertex represents a measured attribute (e.g., gene expression, behavior or genotype), and in which an edge between two attributes is weighted with the

appropriate correlation coefficient. As with microarray data, this graph can be enormous. It is built from the entire trait x trait correlation matrix. A variety of graph algorithms can now be applied, in particular those designed to extract densely-connected subgraphs. By definition the most densely connected subgraphs of all are of course again cliques. A clique in this context may help identify an important biological module (e.g., a subunit of a protein complex). Alternately, it may point to correlation only when a strong driving biological force causes its elements to co-vary. Clique-centric tools provide us with powerful techniques for the study of highly interconnected groups of traits.

We typically seek all maximal cliques², but it can be folly to try to compute them without first knowing a bound on their size. To do this, we solve maximum clique with the aid of fixed-parameter tractability (FPT).

A problem is FPT if it has an algorithm that runs in $O(f(k)n^c)$ time, where n is the problem size, k is the input parameter, and c is a constant independent of both n and k .

Clique is not FPT, however, unless the W hierarchy³ collapses [24]. Thus, we focus instead on clique's complementary dual, the vertex cover problem, and on G' , the complement of G . (G' has the same vertex set as G , but edges present in G are absent in G' and vice versa.) Both clique and vertex cover are \mathcal{NP} -complete. Unlike clique, however, vertex cover is FPT. The relevant observation here is this: a vertex cover of size k in G' turns out to be exactly the complement of a clique of size $n - k$ in G . We therefore search for a minimum vertex cover in G' , thereby finding the desired maximum clique in G . Currently, the fastest known vertex cover algorithm runs in $O(1.2759^k k^{1.5} + kn)$ time [17]. The requisite exponential growth (modulo $\mathcal{P} \neq \mathcal{NP}$) is thus reduced to a mere additive term, making it realistic now to consider the search for cliques of immense sizes. Some of our recent progress on FPT and maximum clique is featured in [12]. Our latest work on high performance solutions to maximal clique enumeration can be found in [39].

4 Noise, Overlap and the Paraclique Method

Clique is an ideal cluster definition, the gold standard. Every vertex (transcript) in a clique must be highly correlated with all others in that clique by definition. Microarray data, on the other hand, at least as generated under current technologies, are inherently noisy. It seems highly unlikely that noise alone could cause an entire clique's worth of correlation coefficients to be excessively high, whereas common clustering methods including KNN and K-Means can allow transcripts into a group that are related due to

² A maximal clique in a graph G is one that is locally optimal. That is, it is a complete subgraph with the property that no new vertex in G can be added to it. It should not be confused with the more common notion of maximum clique, which is a largest clique in G .

³ The W hierarchy, whose lowest level is FPT, can be viewed as a fixed-parameter analog of the polynomial hierarchy, whose lowest level is \mathcal{P} . Such a collapse is widely viewed as an exceedingly unlikely event, roughly on a par with the likelihood of the collapse of the polynomial hierarchy.

noise. Thus, in contrast to many other popular approaches, clique does not seem to be plagued with false positives. On the other hand, if even one coefficient is incorrectly found to be too low, then the clique is lost, equaling a false negative. Several edges may even be missing from the largest subgraphs. The result is that a typical clique analysis may yield exceedingly large numbers of modest-sized, highly-overlapping cliques [31]. To aggregate these data, we want to solve something akin to dense- k -subgraph [25], which is \mathcal{NP} -complete even on graphs of maximum degree three.

To accomplish this we have developed a novel algorithmic approach that we term paraclique. Roughly speaking, a paraclique is a clique augmented with vertices in a highly controlled manner to maintain density. In what follows we describe the paraclique algorithm in pidgin Algol. It uses what we term a glom factor to latch onto new vertices, and an optional threshold to check the original weights of edges discarded by the high pass filter.

```

paraclique (graph  $G$ , glom factor  $g$ , threshold  $t$ )
  set  $P$  to  $C$ , some maximum clique in  $G$ 
  set  $P'$  to  $\emptyset$ 
  while  $P \neq P'$  do
    set  $P'$  to  $P$ 
    for every  $v \in V - P$  do
      if  $v$  is adjacent to at least  $g$  members of  $P'$ 
        then if the weight of each edge connecting  $v$  to  $P'$  before filtering is at least  $t$ 
          then set  $P$  to  $P \cup v$ 
        end for
    end while
  return  $P$ 

```

Although our interest is focused on real and not synthetic data, it is not at all difficult to prove the following.

Theorem. For any graph G , with g set to $|C| - 1$ the edge density of P as computed by the paraclique algorithm is at least 50% as long as $|P| \leq 2|C|$.

Paraclique can be iterated as long as needed, excising from G the current value of P at each pass. Empirical testing on real microarray data has been revealing. If one merely augments a clique with one- and two-neighborhoods, the edge density rapidly falls to the 10 – 20% range. But with paraclique, if we set g to $|C| - 1$, then edge density tends to remain above 90% (even with t set to 0). All this is accomplished while the sizes of the paracliques generated are roughly twice the sizes of the respective cliques from which each was constructed.

5 Sample Results

We have applied our methods to mouse brain gene expression data collected by Robert W. Williams and colleagues and described in [18]. In the genetic analysis of this data,

several major trans-regulatory QTLs were identified using a bottom up approach. Our first application of clique-centric analysis [4] to this data was performed on MAS5.0 normalized brain gene expression data using Spearman's rank correlations. We began with a threshold setting of 0.50 and, using our FPT-based algorithms, consumed roughly a week of CPU time on a 32-processor cluster to determine that the maximum clique size was 369. Analyzing cliques of this size was deemed to be beyond the current capability of effective biological verification methods. We therefore iterated over a variety of thresholds until we settled on a relatively high $|r| \geq 0.85$ value to filter the edge-weighted graph. This analysis revealed 5227 maximal cliques with a maximum clique size of 17. See Figure 2.

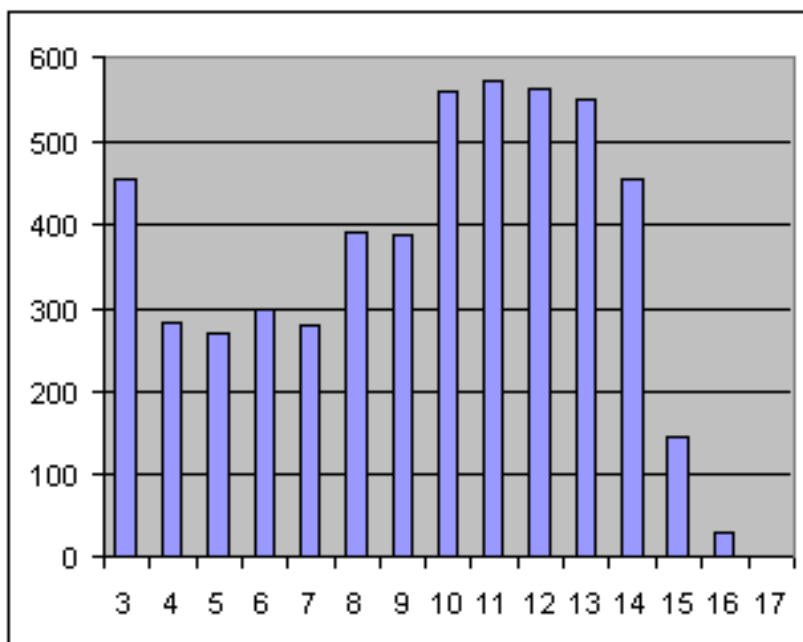


Fig. 2. Clique size distribution in MAS5.0 normalized brain

The gene whose transcript is represented by far in the greatest number of maximal cliques is *Lin7c* (also known as *Veli3*). This gene forms a complex with clique member *Cask* and *Mint1* to participate in the coupling of cell adhesion models to synaptic vesicle cycling [15]. This complex plays a role in the cycling of numerous neurotransmitter receptors, including *Htr2c* [6]. Other *Lin7c* clique members include *Gs2na*, which is a synapse and soma localized gene also involved in protein-protein interactions with consequences on locomotor behavior [5]. For this dataset it is particularly promising that clique-centric analysis has identified groups of genes encoding proteins that physically interact at the synapse. These cliques are highly overlapping. Simple array noise can often be responsible for separating a group of inter-correlated transcripts into a huge

number of highly similar cliques, each varying due to the presence of a small number of missing edges. The annotation and interpretation of such a result is quite challenging. This challenge is even more pronounced with RMA normalizations, because of an overall increase in correlation coefficient values and with it an increase in graph density.

In subsequent analyses, we again studied the aforementioned brain gene expression data. We kept the threshold at 0.85, but used the RMA normalization package due to its greater precision. Correlation graph density is greatly affected by RMA. The maximum clique size increased from 17 to 280; the number of maximal cliques increased from 5227 to a value in excess of 9.5 million (where we stopped counting). Dimensional reduction using paraclique produced highly-purified gene sets, while maintaining densely-connected subgraphs and consolidating the overwhelming number of overlapping cliques. Millions of maximal cliques were reduced to a mere 31 paracliques, ranging in size from 12 to 466, with each paraclique having an edge density in excess of 95%. Thus, as designed, the paraclique algorithm is an attempt to correct for noise. It extracts dense, disjoint subgraphs. By definition, the majority of the interconnections among all transcripts must be present. This does not preclude the presence of edges between paracliques. We have observed that these dense subgraphs are marked by interconnectivity at interfaces between only a few vertices. Thus there are some vertices with high-coverage of multiple paracliques. The corresponding genes are likely to be important players in the regulatory networks with interconnections to these transcripts.

6 Impact

Mapping the QTL regulators of paraclique expression reveals that trans-QTL bands do not act independently, but rather, they act in concert to regulate simultaneously over 1700 transcripts. The paraclique algorithm has given us an unprecedented and simultaneous view of all of transcriptome QTL data. Using a derivative of the cluster map display designed for WebQTL [19], it is possible to visualize the combinations of trans QTL bands that are responsible for regulation of the paracliques. Using paraclique, we were able to decompose the genetic co-expression matrix into groups of transcripts with shared regulatory architecture, and have demonstrated that the trans-bands act in concert, as opposed to singly to regulate transcription. This result could not be obtained using a bottom up approach, which presupposes single regulatory loci, although we did see indications of this structure by simply mapping many functionally related transcripts in parallel. See Figure 3.

Once these key loci are isolated, the challenge is to identify the actual pathways and genes that are involved in biological networks that are perturbed by the genetic polymorphism. A plethora of annotation tools aimed at understanding gene sets have emerged over the past few years. The vast majority of genes that are co-expressed in the paraclique graph are related to the neuronal synapse, and in particular the transport and translation of mRNAs at the dendrites. One of the compelling candidate genes for the regulatory locus at chromosome 1 is *Mtap2*, which is located in the QTL region, and is an expression correlate of transcripts in the region. See Figure 4.

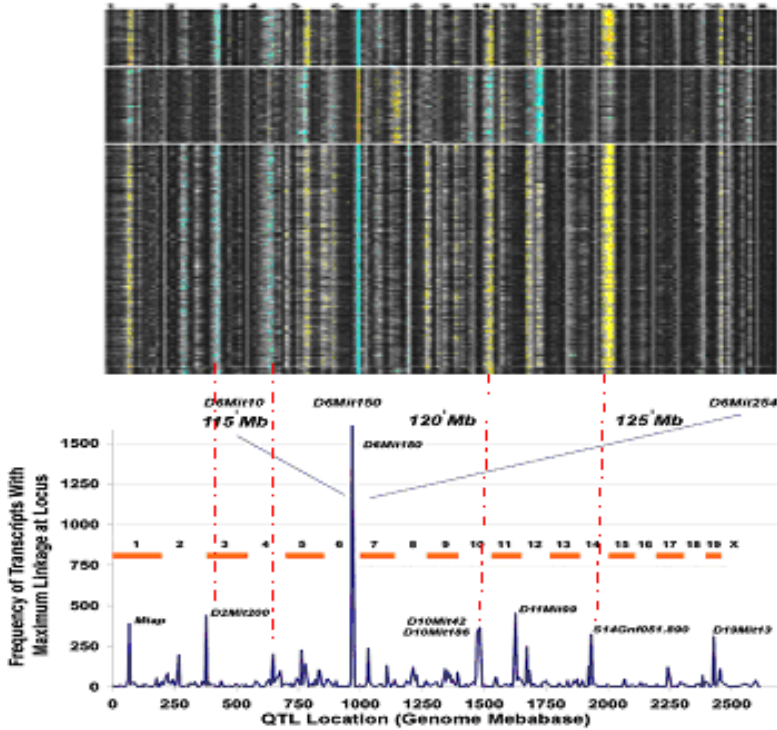


Fig. 3. Trans-QTL bands as shown in the lower panel are compared to the genetic regulation of paracliques as shown in the upper panel. The mouse genome is represented on the x-axis of both plots. The lower plot is courtesy of Nature Genetics [18]. In the upper plot, QTL members of three paracliques are plotted in parallel. Warm colors represent locations where the DBA/2J alleles decrease in expression levels. Cool colors represent locations where C57BL/6J alleles increase in expression levels. Each paraclique is regulated by a unique combination of trans-QTL loci.

7 Candidate Gene Selection

A QTL may contain tens to hundreds of genes, as was illustrated in Figure 3. The identification of candidate genes in a QTL region can be aided by an analysis of multiple converging evidences so as to rank these genes. Let us discuss just two of these evidences.

In the first approach, we integrate simple edge densities with genomic data to identify candidate regulators. We have observed that a small number of vertices within paracliques are highly adjacent to vertices in other paracliques. By examining the percentage of possible adjacent edges observed among paraclique members, we are able to identify the vertices at the interface of two or more paracliques. By further overlaying information regarding the genomic location of the transcripts represented by these vertices and comparing this position to the location of QTLs, we are able to identify paraclique members that contain the causative polymorphisms responsible for the covariance of paracliques. This approach is not exclusionary, nor is it fully inclusionary. This is because expression data is not available for some genes in the QTL interval. Moreover, in

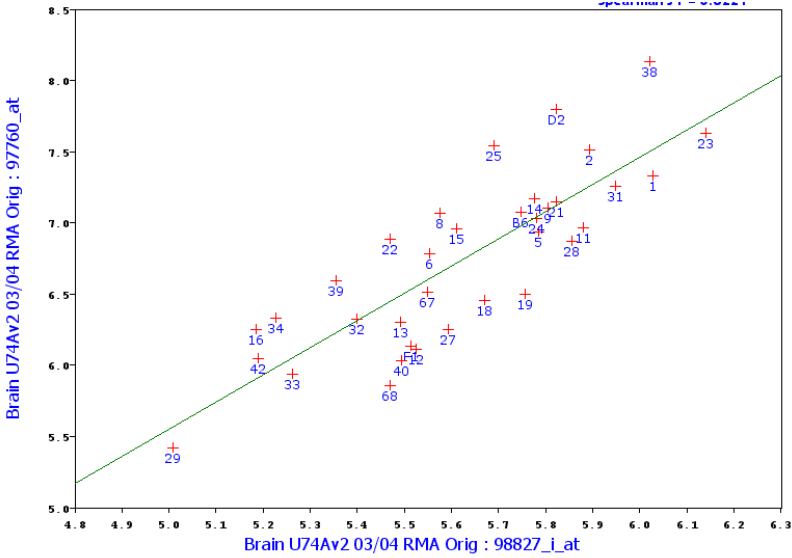


Fig. 4. An example of genetic correlation scatter plot revealing a high correlation between *Kif5a* and *Mtap2* abundance

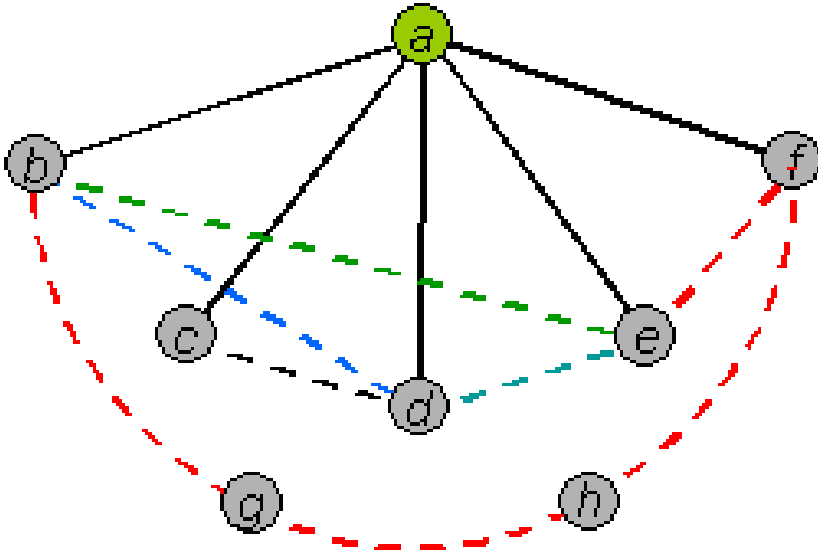


Fig. 5. A phenotype-centered gene neighborhood graph. Vertex connectivity is used to identify genes that are more robust to mutation. Vertex *d* represents a gene that is in the neighborhood of the phenotype of interest, *a*, based on genetic correlations in a reference population. Because *d* is a member of several vertex disjoint paths between *b* and *e*, it is more robust to perturbations than other genes in the network.

other cases, the polymorphism has functional effects on the gene but does not influence transcript abundance.

In the second approach, we borrow a page from evolutionary theory, which suggests that genes residing in one of many redundant paths are more mutable than those that are in exclusive pathways [38]. Suppose vertex a is a transcript abundance or higher order phenotype, and suppose genes b, c, d, e and f are the neighbors of a (determined by genetic correlation of transcript abundance to this phenotype in RI lines). We seek to find redundant pathways, aka vertex-disjoint paths, in a 's gene neighborhood. See Figure 5.

Consider, for example, genes b and e . They may be adjacent via a path of length one, as shown in green. Alternately, they may be connected by one or more paths of length two or more within a 's neighborhood, as shown in blue. In fact they may even be connected by one or more paths of length two or more via genes not in a 's neighborhood, as shown in red. Because one example of each type is depicted in Figure 5, and because these paths are vertex disjoint, we would in this case say that b and e are 3-connected. The biological interpretation is that three distinct pathways join b and e , thereby providing redundancy, around d . Thus, mutations in d are likely to be accommodated by other alternate pathways. We will use a min-max characterization to score a gene in the neighborhood of a phenotype. To illustrate, let x, y and z denote three such genes with $x \neq y \neq z$. The score of x relative to y and z , denoted $R_{yz}(x)$, is the maximum number of vertex-disjoint paths, exclusive of a , that remain to connect y and z if x is eliminated from the phenotype-centered-gene-neighborhood graph. The robustness score of x , denoted $R(x)$, is the minimum value of $R_{yz}(x)$ taken over all pairs y and z for which x lies on some path between y and z . In figure 5, for example, $R_{be}(d) = 2$, because the elimination of d leaves the two paths $b - e$ and $b - g - h - f - e$. But all paths between c and e must go through d , and so $R_{ce}(d) = R(d) = 0$.

8 Vertex Coverage and Gene-to-Phenotype Networks

Determining the relationship between higher order phenotypes and paracliques and other dense subgraphs in the genetic co-expression network is made readily possible using the reference population. Over 1500 phenotypes have been obtained in mouse recombinant inbred strains. The phenotypes are collected in various subsets of the strain population, and unlike the gene expression correlations, may contain substantial missing in a non-uniform pattern across the data matrix. We have thus used p -value for the genetic correlation as an edge weight threshold. An example is shown in Figure 6, in which a small paraclique of brain transcript abundances containing an inwardly rectifying voltage-gated potassium channel, *Kcnj9*, is heavily involved in behavioral variation and shown to be related to blood alcohol at the return of the righting reflex and preference for sweet solutions.

9 Scalability, Data Dimensionality, and Directions for Future Research

An exciting potential of systems genetics is continually to aggregate biological data in reference populations across all levels of biological scale in a highly complex multicellular organism, the laboratory mouse. Already, there are massive numbers of unique

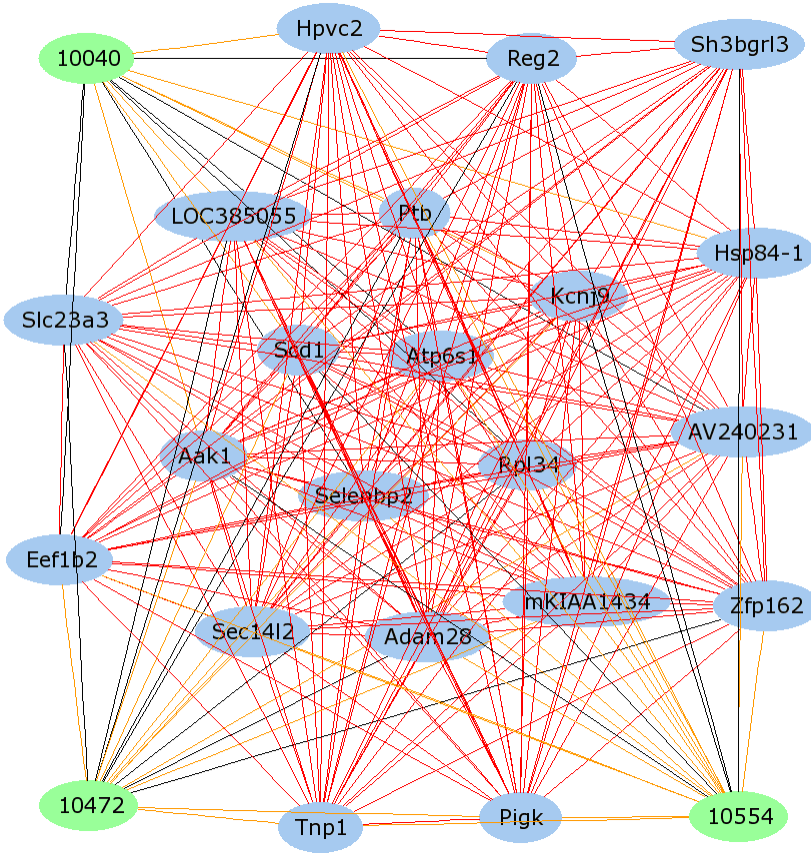


Fig. 6. At an edge weight threshold of $p < .05$, the phenotypes 10472 and 10554, shown in green, cover 100% of the paraclique members. Phenotype 10040 covers 95.42% of the paraclique members. Phenotype vertex labels are as follows: 10040 - BEC at return of Righting Reflex 10472, 10554 - Saccharin preference basal and after EtOH.

genotype vectors in several existing reference populations, and expression assays for thousands of genes in at least ten tissue types. At present, there are approximately 1500 higher order phenotypes available in these populations, and many more in collection. As the reliability and precision of high-throughput proteomics and cell-culture assays improves, the amount of data available in these lines will increase markedly. Additional data will produce multiplicative growth in the number of correlations that are defined. While linear modeling and other parametric approaches have been applied with much success for known pathways, extracting novel information from data of this scale is a phenomenal challenge. Discretizing this complex correlational system and applying advances in graph algorithms have given us an efficient and relatively rapid means for reducing the data dimensionality and extracting networks of genes and phenotypes. The approaches that we have illustrated are highly scalable, and are capable of extracting large sets of related traits from the entire relational system.

In addition to expansive volumes of data, there is a growing complexity to the types of research questions that can be asked. We are presently developing approaches to compare graphs collected in a systems genetic context to reflect differences in time, tissue and treatment effects. Visualization methods and compelling biological validation of novel results are essential to translate these methods and deliver them to the broader audience of biologists who are already successfully harnessing the insight into specific gene-regulatory relations that these public data sets have allowed.

Acknowledgments

This work has been supported in part by the National Science Foundation under grant CCR-0311500, by the Office of Naval Research under grant N00014-01-1-0608, and by the National Institutes of Health under grants 1-P01-DA-015027-01 and 1-R01-MH-074460-01. It has employed resources managed by UT-Battelle for the U.S. Department of Energy under contract DE-AC05-00OR22725. We thank Dr. Robert W. Williams and Dr. Lu Lu of the University of Tennessee Health Science Center for providing data for the analyses we have described here, and Evan G. Williams for rendering Figure 3.

References

1. F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proceedings, Workshop on Algorithm Engineering and Experiments*, New Orleans, Louisiana, 2004.
2. F. N. Abu-Khzam, M. A. Langston, P. Shanbhag, and C. T. Symons. Scalable parallel algorithms for FPT problems. *Algorithmica*, 2006, accepted for publication.
3. O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97:10101–10106, 2000.
4. N. E. Baldwin, E. J. Chesler, S. Kirov, M. A. Langston, J. R. Snoddy, R. W. Williams, and B. Zhang. Computational, integrative, and comparative methods for the elucidation of genetic coexpression networks. *Journal of Biomedicine and Biotechnology*, 2:172–180, 2005.
5. M. Bartoli, J. P. Ternaux, C. Forni, P. Portalier, P. Salin, M. Amalric, and A. Monneron. Down-regulation of striatin, a neuronal calmodulin-binding protein, impairs rat locomotor activity. *Journal of Neurobiology*, 40:234–243, 1999.
6. C. Becamel, S. Gavarini, B. Chanrion, G. Alonso, N. Galeotti, A. Dumuis, J. Bockaert, and P. Marin. The serotonin 5-ht_{2a} and 5-ht_{2c} receptors interact with specific sets of pdz proteins. *Journal of Biological Chemistry*, 279:20257–20266, 2004.
7. A. Bellaachia, D. Portnoy, Y. Chen, and A. G. Elkahoul. E-cast: A data mining algorithm for gene expression data. In *Proceedings, Workshop on Data Mining in Bioinformatics*, Edmonton, Alberta, Canada, 2002.
8. A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, pages 54–64, 2000.
9. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

10. I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. The maximum clique problem. In D. Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 4. Kluwer Academic Publishers, 1999.
11. R. B. Brem and L. Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences*, 102:1572–1577, 2005.
12. R. B. Brem, G. Yvert, R. Clinton, and L. Kruglyak. Genetic dissection of transcriptional regulation in budding yeast. *Science*, 296:752–755, 2002.
13. K. W. Broman and T. P. Speed. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64:641–656, 2002.
14. A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, and I. S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97:12182–12186, 2000.
15. S. Butz, M. Okamoto, and T. C. Sudhof. A tripartite protein complex with the potential to couple synaptic vesicle exocytosis to cell adhesion in brain. *Cell*, 94:773–782, 1998.
16. L. Bystrykh, E. Weersing, B. Dontje, S. Sutton, M. T. Pletcher, T. Wiltshire, A. Su, E. Veltinga, J. Wang, K. F. Manly, L. Lu, E. J. Chesler, R. Alberts, R. C. Jansen, R. W. Williams, M. P. Cooke, and G. d. Haan. Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nature Genetics*, 37:225–232, 2005.
17. L. S. Chandran and F. Grandoni. Refined memorisation for vertex cover. In *Proceedings, International Workshop on Parameterized and Exact Computation (IWPEC)*, 2004.
18. E. J. Chesler, L. Lu, S. Shou, Y. Qu, J. Gu, J. Wang, H. C. Hsu, J. D. Mountz, N. E. Baldwin, M. A. Langston, J. B. Hogenesch, D. W. Threadgill, K. F. Manly, and R. W. Williams. Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, 37:233–242, 2005.
19. E. J. Chesler, L. Lu, J. Wang, R. W. Williams, and K. F. Manly. Webqtl: Rapid exploratory analysis of gene expression and genetic networks for brain and behavior. *Nature Neuroscience*, 7:486–486, 2004.
20. E. J. Chesler, J. Wang, L. Lu, Y. Qu, K. F. Manly, and R. W. Williams. Genetic correlates of gene expression in recombinant inbred strains: a relational model system to explore neurobehavioral phenotypes. *Neuroinformatics*, 1:343–357, 2003.
21. E. J. Chesler and R. W. Williams. Brain gene expression: Genomics and genetics. *International Review of Neurobiology*, 60:59–95, 2004.
22. G. A. Churchill, D. C. Airey, H. Allayee, J. M. Angel, A. D. Attie, J. Beatty, W. D. Beavis, J. K. Belknap, B. Bennett, W. Berrettini, A. Bleich, M. Bogue, K. W. Broman, K. J. Buck, E. Buckler, M. Burmeister, E. J. Chesler, J. M. Cheverud, S. Clapcote, M. N. Cook, R. D. Cox, J. C. Crabbe, W. E. Crusio, A. Darvasi, C. F. Deschepper, R. W. Doerge, C. R. Farber, J. Forejt, D. Gaile, S. J. Garlow, H. Geiger, H. Gershenfeld, T. Gordon, J. Gu, W. Gu, G. d. Haan, N. L. Hayes, C. Heller, H. Himmelbauer, R. Hitzemann, K. Hunter, H. C. Hsu, F. A. Iraqi, B. Ivandic, H. J. Jacob, R. C. Jansen, K. J. Jepsen, D. K. Johnson, T. E. Johnson, G. Kempermann, C. Kendzierski, M. Kotb, R. F. Kooy, B. Llamas, F. Lammert, J. M. Lassalle, P. R. Lowenstein, A. L. L. Lu, K. F. Manly, R. Marcucio, D. Matthews, J. F. Medrano, D. R. Miller, G. Mittleman, B. A. Mock, J. S. Mogil, X. Montagutelli, G. Morahan, D. G. Morris, R. Mott, J. H. Nadeau, H. Nagase, R. S. Nowakowski, B. F. O'Hara, A. V. Osadchuk, G. P. Page, A. Paigen, K. Paigen, A. A. Palmer, H. J. Pan, L. Peltonen-Palotie, J. Peirce, D. Pomp, M. Pravenec, D. R. Prows, Z. Qi, R. H. Reeves, J. Roder, G. D. Rosen, E. E. Schadt, L. C. Schalkwyk, Z. Seltzer, K. Shimomura, S. Shou, M. J. Sillanpaa, L. D. Siracusa, H. W. Snoeck, J. L. Spearow, K. Svenson, L. M. Tarantino, D. Threadgill, L. A. Toth, W. Valdar, F. P. d. Villena, C. Warden, S. Whatley, R. W. Williams, T. Wiltshire, N. Yi, D. Zhang, M. Zhang, and F. Zou. The collaborative cross, a community resource for the genetic analysis of complex traits. *Nature Genetics*, 36:1133–1137, 2004.

23. R. W. Doerge. Mapping and analysis of quantitative trait loci in experimental populations. *Nature Reviews Genetics*, 3:43–52, 2002.
24. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
25. U. Feige, D. Peleg, and G. Kortsarz. The dense k -subgraph problem. *Algorithmica*, 29:410–421, 2001.
26. M. Girolami and R. Breitling. Biologically valid linear factor models of gene expression. *Bioinformatics*, 20:3021–3033, 2004.
27. P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.
28. E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrachs, and R. Shamir. An algorithm for clustering cDNAs for gene expression analysis. In *Proceedings, RECOMB*, Lyon, France, 1999.
29. L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.
30. N. Hubner, C. A. Wallace, H. Zimdahl, E. Petretto, H. Schulz, F. Maciver, M. Mueller, O. Hummel, J. Monti, V. Zidek, A. Musilova, V. Kren, H. Causton, L. Game, G. Born, S. Schmidt, A. Muller, S. A. Cook, T. W. Kurtz, J. Whittaker, M. Pravenec, and T. J. Aitman. Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nature Genetics*, 37:243–253, 2005.
31. M. A. Langston, L. Lan, X. Peng, N. E. Baldwin, C. T. Symons, B. Zhang, and J. R. Snoddy. A combinatorial approach to the analysis of differential gene expression data: The use of graph algorithms for disease prediction and screening. In J. S. Shoemaker and S. M. Lin, editors, *Methods of Microarray Data Analysis IV*. Springer Verlag, 2005.
32. M. A. Langston, A. D. Perkins, A. M. Saxton, J. A. Scharff, and B. H. Voy. Innovative computational methods for transcriptomic data analysis. In *Proceedings, ACM Symposium on Applied Computing*, Dijon, France, 2006, accepted for publication.
33. J. Li and M. Burmeister. Genetical genomics: Combining genetics with gene expression analysis. *Human Molecular Genetics*, 14:163–169, 2005.
34. K. F. Manly and J. M. Olson. Overview of qtl mapping software and introduction to map manager qt. *Mammalian Genome*, 10:327–334, 1999.
35. J. L. Peirce, L. Lu, J. Gu, L. M. Silver, and R. W. Williams. A new set of bxd recombinant inbred lines from advanced intercross populations in mice. *BMC Genetics*, 5:7, 2004.
36. E. E. Schadt, S. A. Monks, T. A. Drake, A. J. Luskis, N. Che, V. Colinao, T. G. Ruff, S. B. Milligan, J. R. Lamb, G. Cavet, P. S. Linsley, M. Mao, R. B. Stoughton, and S. H. Friend. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422:297–302, 2003.
37. D. K. Slonim. From patterns to pathways: gene expression data analysis comes of age. *Nature*, 32:502–508, 2002.
38. A. Wagner. Distributed robustness versus redundancy as causes of mutational robustness. *Bioessays*, 27:176–188, 2005.
39. Y. Zhang, F. N. Abu-Khzam, N. E. Baldwin, E. J. Chesler, M. A. Langston, and N. F. Samatova. Genome-scale computational approaches to memory-intensive applications in systems biology. In *Proceedings, Supercomputing*, Seattle, Washington, 2005.

Topological Robustness of the Protein-Protein Interaction Networks

Chien-Hung Huang¹, Jywe-Fei Fang², Jeffrey J.P. Tsai^{3,4}, and Ka-Lok Ng^{4,*}

¹ Department of Computer Science and Information Engineering, National Formosa University, 64, Wen-Hwa Road, Hu-wei, Yun-Lin, Taiwan 632

² Department of Digital Content and Technology, National Taichung University, 140 Min-Shen Rd. Taichung, Taiwan 403

³ Department of Computer Science, University of Illinois, 851 S. Morgan (M/C 152), Room 1120 SEO, Chicago, IL 60607-7053, USA

⁴ Department of Biotechnology and Bioinformatics, Asia University, No. 500, Lioufeng Road, Wufeng Shiang, Taichung, Taiwan 413
klng@asia.edu.tw

Abstract. The stability and fragility of four species' protein-protein interaction networks (PINs) are studied by investigating their robustness, i.e. their topological parameters retain a similar system behavior with respect to four different types of perturbations. Four types of perturbations are considered; that is (i) network nodes are randomly removed (failure), (ii) the most connected node is successively removed (attack), (iii) interaction edges are rewired randomly, and (iv) edges are randomly deleted. At most 50% of network nodes or edges are deleted or rewired. It is demonstrated that PINs are quite robust with respect to failure, attack, random rewiring and edge deletion, that is the average diameters for perturbed networks differ from the unperturbed cases have a difference less than 13%, which is relative small in comparison with WWW and the Internet results. These results suggest that PINs' network topologies are robust with respect to perturbations.

1 Introduction

The increasing availability of biological data (protein-protein interactions, metabolic and gene transcription) suggest that many biological networks possess the scale-free property [1-3], and they have attracted much attention in recent years. Similarly in many real life networks, such as World-Wide Web [4,5], the Internet [6], movie actor collaboration network [7], and social network [8], they are also shown to display scale-free properties.

The question of the stability of a biological network upon perturbation raised a lot of interests in the last few years, such as biochemical networks [9] and the class of scale-free bio-logical networks. Albert et al. [10] demonstrated that the scale-free network display a surprising degree of tolerance against errors, in other words, scale-free networks is very robust against perturbation. Two types of perturbations were considered in their study; the first one was to remove nodes randomly (so-called

* Corresponding author.

failure), and the second one was to remove the most connected node, and continue selecting and removing nodes in decreasing order of their connectivity (so-called attack). In fact, it was demonstrated that scale-free network is robust against the first type of perturbation only and not the second one. Albert et al. [10] studied the changes in average network diameter when a small fraction of the nodes is removed (failure and attack) for WWW and the Internet with different sizes (1,000, 5,000 and 20,000 nodes). It was found that scale-free networks are robust against random failures, a property not shared by their random counterparts. However, the average network diameter of a scale-free network increases rapidly with respect to attack.

Later, Jeong et al. [2] applied the same error tolerance technique to study error tolerance of the protein-protein interaction network (PIN) in yeast cells, and showed that the connectivity of a protein in the network is correlated with the likelihood that its removal could be lethal to the cell. Another piece of work [11] applied the error tolerance idea to study the complexity and fragility of three ecosystem food webs, and it was shown that ecological networks are very robust against random removals but could be extremely fragile when selective attack were used. These pieces of works had shed some light and open the door to study the stability of many biological systems. Therefore, it is meaningful to do a more complete analysis, and study how the robustness arises from the underlying organization or structure of the network.

In this paper, we perform a throughout *in silo* study of the biological network stability problem, by considering four different types of perturbations in our simulation. Each type of perturbation corresponds to different interpretation or mechanism, which is described in the methods section. In specific, we study those effects for PINs, since most of the cellular processes are the results of a cascade of events mediated by protein-protein interactions.

The protein-protein interaction database DIP [12] was used as our input data. A throughout error tolerance study of PINs for four species (*H. pylori*, *C. elegans*, *D. melanogaster*, and *S. cerevisiae*) was performed. There are several motivations why we want to extend the previous works. First, so far only the yeast PIN was studied [2], and to the best of our knowledge no comparison was made with other species. Second, only two types of perturbations were considered [10], and we considered two more different types of perturbations in this work, that is a total of four. Third, there were concerns that large fraction of the protein-protein interaction data are false negative and false positive signals. In order to test this point, we perturbed PINs with a much higher error rates (up to 50%, where only a few percents was considered in Ref. 10).

2 Method

The protein-protein interactions database DIP was employed as our input. DIP is a database that documents experimentally determined protein-protein interactions for seven species, that is, *E. coli*, *H. pylori*, *C. elegans*, *D. melanogaster*, *H. sapiens*, *M. musculus* and *S. cerevisiae*.

The random graph theory approach [13, 14] was employed in this work to study PINs. In the graph theory approach, each protein is represented as a node and interaction as an edge. By analyzing the DIP database one constructed an adjacency matrix to represent the PIN. In the adjacency matrix a value of zero, one and infinity is assigned to represent self-interacting, direct interacting and non-interacting protein

respectively. The PINs discussed above have complex topology. A complex network can be characterized by certain topological measurements, such as node degrees, node degree distributions, average network diameters and node degree correlation functions [15]. For instance, one of the topological features of a complex network is its node degree distribution. From the adjacency matrix, one can obtain a histogram of k interactions for each node. Dividing each point of the histogram with the total number of proteins provide $P(k)$. For scale-free network, the degree distribution has a power-law distribution, $P(k) \sim k^{-\gamma}$, where γ is a constant. The power-law form of the degree distribution implies that the network is extremely inhomogeneous. We had performed a global topological study of PIN for seven species [15] and demonstrated that to a good approximation they all belong to the class of scale-free networks.

One of the major problems of using the DIP database data was that it had both false-negative and false-positive results [16] arising from the yeast two-hybrid (Y2H) screening method [17]. The Y2H method was known to have a number of limitations, such as the possibility of unexpected trans-activation by the protein linked to the DNA-binding domain, and it can only be used to identify interactions that can be recreated within the environment of the yeast.

In Maslov and Sneppen paper [18], the authors argued that one could take into account of experimental artifacts by doing an error tolerance study, and it was showed that network topological results, such as the node degree correlation profile, were quite robust to experimental errors. In this paper, we consider four types of perturbations; (i) network nodes are removed randomly (failure), (ii) the most connected node is successively removed (attack), (iii) nodes are rewired randomly, the so-called local rewiring algorithm [19-21], in which this process does not change the node degrees of each node, and (iv) edges are deleted randomly in which this process does not change the total number of nodes of the network.

For the first type of perturbation (failure), one can interpret that it is equivalent to introducing false negative data (that is the remove protein has no interaction with other proteins). On the other hand, this perturbation can also be interpreted as a malfunctioning of the protein (due to mutation) or simulating the gene knockout experiment.

For the second type of perturbation (attack), it is equivalent to remove a hub and we expect that this can alter the network's topology [10]. This perturbation can be interpreted as a malfunctioning of the most highly interacted protein first and continue selecting and removing nodes in decreasing order of their connectivity.

For the third type of perturbation, nodes are rewired randomly. First, we randomly selected a pair of edges A-B and C-D. The two edges are then rewired in such a way that A connects to D, while C connects to B. Notice that this process does not change the node degrees of each node. This perturbation can be interpreted as introducing both false positive and false negative interactions into the network while keeping the total number of nodes unchanged. The above rewiring steps are repeated, this process leads to a randomized perturbation of the original network. The main reason we consider this perturbation is because there are concerns [22] that the experimental bias of the yeast two-hybrid experiment could possibly spoil topological parameters calculation, such as the average network diameter calculations. Therefore, in order to test whether the interaction network is robust against errors, we perturb the networks by random rewiring. Furthermore, Mering et al. [23] suggested that 90% of the protein-protein interaction data of yeast were false negative, while there were 50% false positive signals, therefore, we tested this concern for four species' networks with a high

error rates (up to 50% of the total numbers of nodes are rewired), since there is no reason to assume that interaction data of different organisms have lower error rates.

For the fourth type of perturbation (edge deletion), one can interpret that it is equivalent to introducing false negative data (that is remove connected nodes) into the networks. Note that this process different from the failure perturbation in that it does not change the total number of nodes in the network.

As more and more nodes or edges are removed, the distance between two nodes tends to increase, impeding the transmission of protein-protein interaction information. The larger the distance is the longer is the expected interaction path length. It seems natural to use the average network diameter to measure the effect of nodes or edges removal. The effects of those perturbations can be characterized by computing the average of the shortest path lengths over all pairs of nodes. For all pairs of proteins, the shortest interaction path length L (i.e. the smallest number of interactions by which one can reach protein 2 from protein 1) is determined by using the Floyd algorithm [24].

We repeat the above perturbation simulation for a fraction f of the total number of nodes or edges, and leads to a randomized perturbation of the original network. The perturbed results are compared with the unperturbed network diameter, i.e. $\Delta_X = (d_X - d)/d * 100\%$ where X stands for *fail*, *att*, *rew* or *edge*. Multiple sampling of the randomized networks allow us to calculate the ensemble average diameters of the perturbed networks, that is $\langle d_{fail} \rangle$, $\langle d_{att} \rangle$, $\langle d_{rew} \rangle$ and $\langle d_{edge} \rangle$ for the four types of perturbations. Network diameter d_X is given by the average of the shortest path lengths over all pairs of nodes,

$$d_X = \frac{\sum_L L f(L)}{\sum_L f(L)} \quad (1)$$

where L is the shortest path length and $f(L)$ is the number of nodes have a path length of L .

The PINs are analyzed using graph theory. Table 1 summarizes the DIP statistics of the seven species; the total number of proteins N_p , total number of interactions N_E , total number of proteins in the largest cluster $N_{largest}$, and the ratio of the size of the largest cluster to the total number of proteins $N_{largest}/N_p$.

Due to the data availability, some of the species' PIN reconstruct from the data could have many isolated components, therefore, in order to obtain reliable results we consider networks with at least 90% of the nodes are connected, which is valid for *C. elegans*, *D. melanogaster*, *H. pylori* and *S. cerevisiae*.

In Figure 1, the plots of $\langle d_{fail} \rangle$ and $\langle d_{edge} \rangle$ vs. the fraction of change for failure and edge deletion perturbations are depicted respectively. It is evident from the figure that both $\langle d_{fail} \rangle$ and $\langle d_{edge} \rangle$ increase with f . This is due to fact that the average path length of any two proteins increases as more and more proteins are removed from the network. It is also noted that almost all the $\langle d_{fail} \rangle$ values are larger than $\langle d_{edge} \rangle$ for the four species regardless of f , the exceptions could possibly due to fluctuations in simulation. One could understand this finding simply because proteins could have multiple interactions ($k > 1$), hence, removing a single protein from the network could have a stronger effect than just an edge deletion.

Table 1. A summary of the vital statistics of the seven species' PINs, the total number of proteins N_p , total number of interactions N_E , total number of proteins in the largest cluster $N_{largest}$, and the ratio of the size of the largest cluster to the total number of proteins $N_{largest}/N_p$

Organism	N_p	N_E	$N_{largest}$	$N_{largest}/N_p$
<i>E. coli</i>	336	611	145	0.43
<i>H. pylori</i>	702	1420	686	0.98
<i>C. elegans</i>	2629	4030	2386	0.91
<i>D. melanogaster</i>	7057	20988	6926	0.98
<i>H. sapiens</i>	1059	1369	563	0.53
<i>M. musculus</i>	327	286	49	0.15
<i>S. cerevisiae</i>	2609	6574	2440	0.94

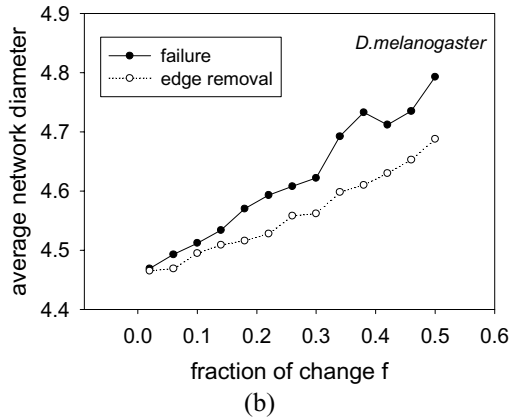
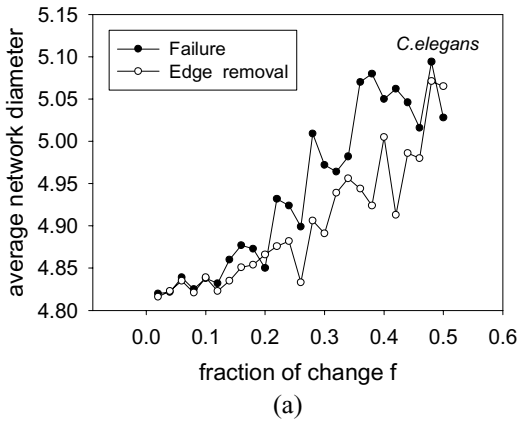


Fig. 1. The average network diameter $\langle d_{fail} \rangle$ and $\langle d_{edge} \rangle$ vs. the fraction of change f for (a) *C. elegans* (b) *D. melanogaster* (c) *H. pylori* and (d) *S. cerevisiae*

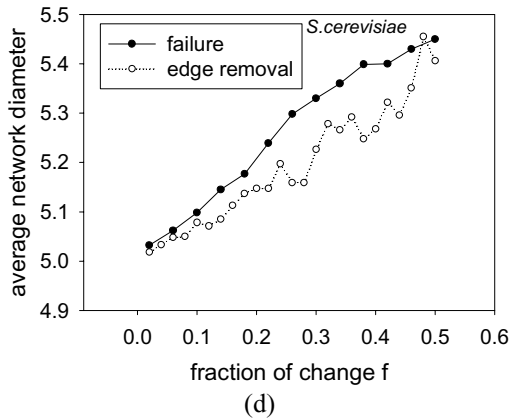
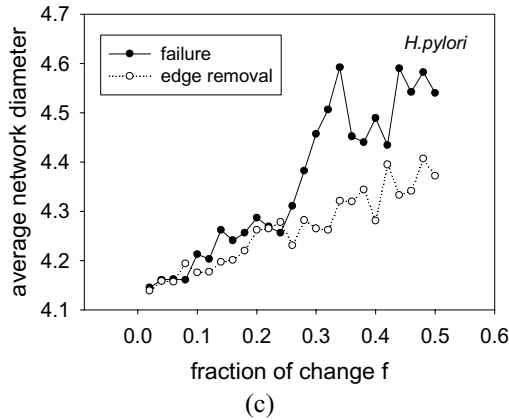
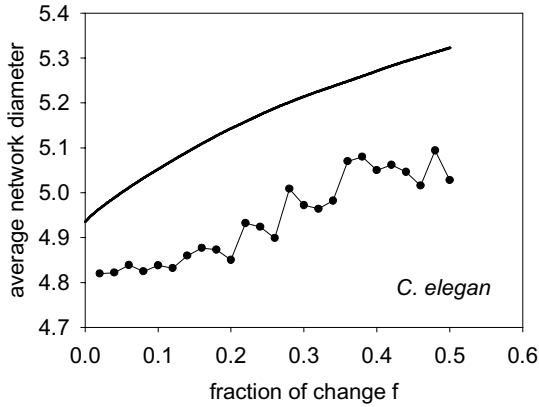


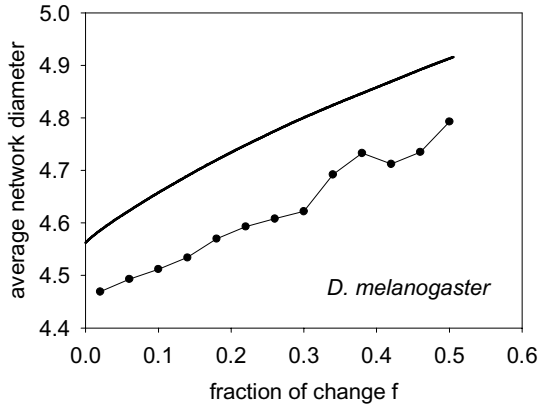
Fig. 1. (continued)

In Figure 2, the plots of $\langle d_{fail} \rangle$ (the see-saw line) and $\langle d_{att} \rangle$ (the solid line) vs. the fraction of removed nodes is depicted. It is evident from Figures 2(a)-(d) that $\langle d_{att} \rangle$ increase monotonically with f and they have values greater than $\langle d_{fail} \rangle$ regardless of f ; that is PINs are vulnerable to attack. One can interpret these results due to the fact that hubs removal could cause a much larger effects than removing nodes randomly.

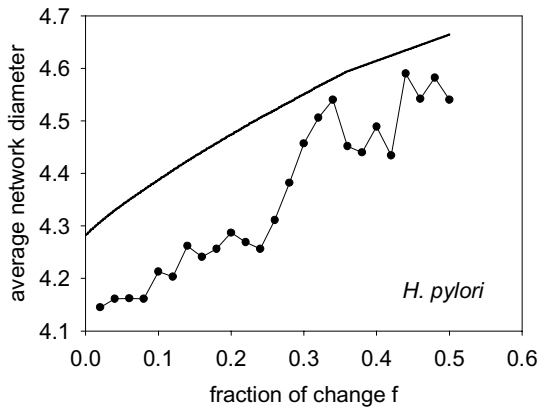
In Figure 3, the plots of $\langle d_{rew} \rangle$ vs. fraction of rewired edges are depicted. It is evident from Figures 3(a)-(d) that the average network diameters $\langle d_{rew} \rangle$ decrease with f . These results are due to the fact that every two pairs of nodes that are selected and rewire are distinct pairs, so nodes that are far apart could possibly connected, hence reducing the average network diameters of the networks. It is also noted the plots for fruit fly in Figures 1, 2 and 3 are relative smoother, these could possibly due to the fact that the fly has a larger dataset (for logarithmic size correction, see [10]).



(a)



(b)



(c)

Fig. 2. The average network diameter $\langle d_{fail} \rangle$ (the see-saw line) and $\langle d_{att} \rangle$ (the solid line) vs. the fraction of change f (a) *C. elegans* (b) *D. melanogaster* (c) *H. pylori* and (d) *S. cerevisiae*

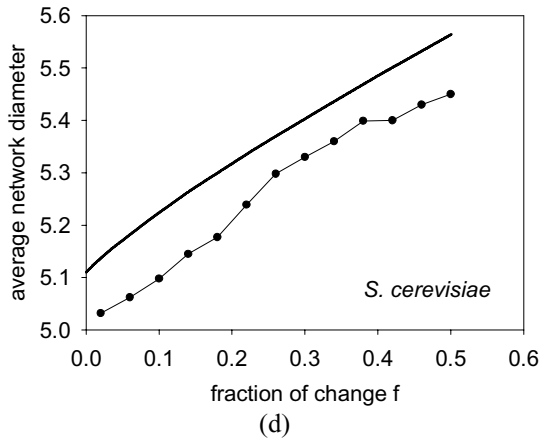


Fig. 2. (continued)

In Table 2, the values of $\langle d_{fail} \rangle$, $\langle d_{att} \rangle$, $\langle d_{rew} \rangle$ and $\langle d_{edge} \rangle$ for the four species at the 50% level of changes are summarized.

In Table 3, the values of the percentage changes of $\langle d_{fail} \rangle$, $\langle d_{att} \rangle$, $\langle d_{rew} \rangle$ and $\langle d_{edge} \rangle$ for the four species at the 50% level of changes are summarized. It is found that the absolute values of the maximum percentage changes $|\Delta_X|$ are less than 13% for the four types of perturbations. Generally speaking, removal of the most connected nodes (attack) has a larger effect on network diameter change. On the other hand, Albert and Barabasi found that [10] diameters of WWW and the Internet doubled its original value if 5% of the most connected nodes were removed. In contrast, the change in diameter is less drastic for PINs with respect to attack. Nevertheless, attack induced a larger effect on average network diameter relative to failure or edge deletion perturbation. The situation is somewhat inconclusive for rewiring perturbation because it could also induce comparable percentage change as attack, i.e. the Δ_{att} and Δ_{rew} columns in Table 3.

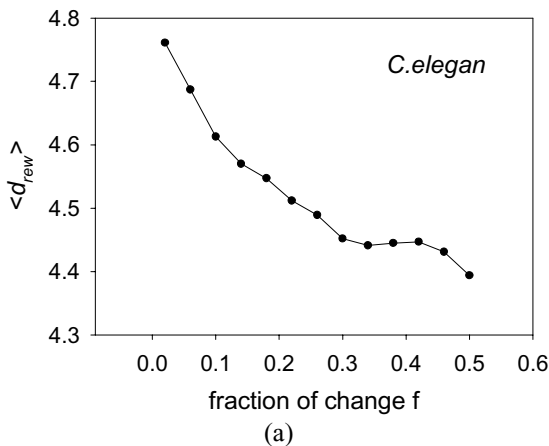


Fig. 3. The average network diameter $\langle d_{rew} \rangle$ vs. the fraction of change of rewired edges for (a) *C. elegans* (b) *D. melanogaster* (c) *H. pylori* and (d) *S. cerevisiae*

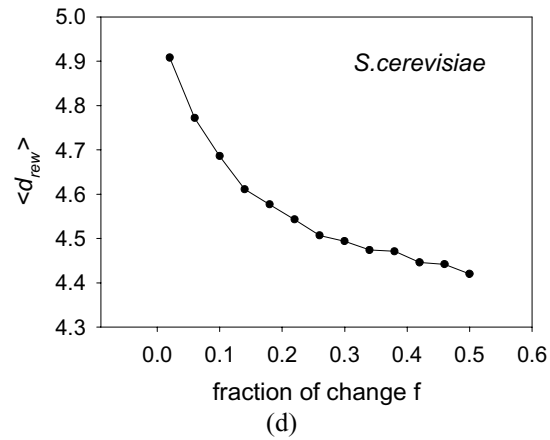
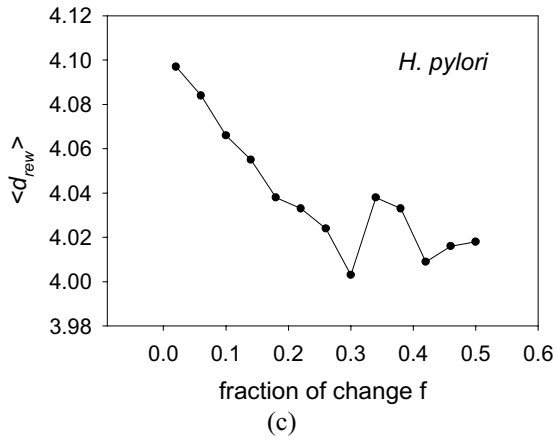
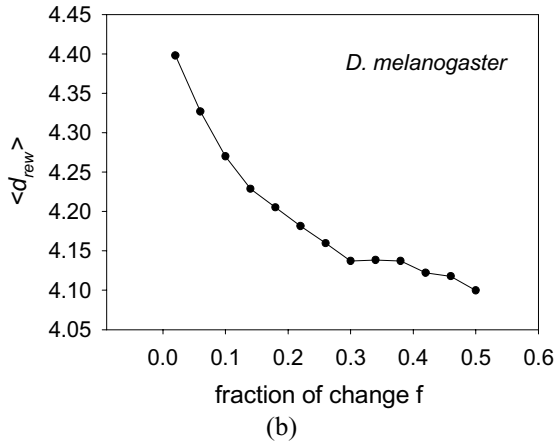


Fig. 3. (continued)

Table 2. A summary of average network diameter d , average failure diameter $\langle d_{fail} \rangle$, average attack diameter $\langle d_{att} \rangle$, average rewire diameter $\langle d_{rew} \rangle$, and average edge deletion diameter $\langle d_{edge} \rangle$ results for the four species at the 50% level of changes

Species	d	$\langle d_{fail} \rangle$	$\langle d_{att} \rangle$	$\langle d_{rew} \rangle$	$\langle d_{edge} \rangle$
<i>H. pylori</i>	4.14	4.54	4.67	4.02	4.37
<i>C.elegans</i>	4.81	5.09	5.32	4.39	5.07
<i>D. melanogaster</i>	4.46	4.79	4.91	4.01	4.69
<i>S. cerevisiae(CORE)</i>	5.01	5.45	5.56	4.42	5.41

Table 3. A summary of the percentage changes for average failure diameter Δ_{fail} , average attack diameter Δ_{att} , average rewire diameter Δ_{rew} , and average edge deletion diameter Δ_{edge} results for the four species at the 50% level of changes

Species	Δ_{fail}	Δ_{att}	Δ_{rew}	Δ_{edge}
<i>H. pylori</i>	9.7	12.8	-2.9	5.6
<i>C.elegans</i>	5.8	10.6	-8.7	5.4
<i>D. melanogaster</i>	7.4	10.1	-10.1	5.2
<i>S. cerevisiae(CORE)</i>	8.8	11.0	-11.8	8.0

We conclude from Table 3 that PINs are quite robust for the four types of perturbations, since $|\Delta_x| < 13\%$ for all the perturbed cases at the 50% level of changes. These results also imply that the average network diameter calculation is robust with respect to false negative and false positive experimental errors.

3 Discussion and Conclusion

We introduce four types of perturbations and consider their effects on PINs for four different species (*H. pylori*, *C. elegans*, *D. melanogaster*, and *S. cerevisiae*). The topological parameter average network diameter was employed to characterize PINs. It is demonstrated that PINs are quite robust with respect to failure, attack, random rewiring and edge deletion perturbations, that is the average network diameters for the perturbed cases, $\langle d_{fail} \rangle$, $\langle d_{att} \rangle$, $\langle d_{rew} \rangle$ and $\langle d_{edge} \rangle$ differed from unperturbed cases with $|\Delta_x| < 13\%$ at a 50% level of changes. Given these results one could argue that network topology results are robust against experimental artifacts, i.e. the false negative and false positive errors.

As more and more nodes are removed from a PIN, clusters of nodes whose links to the system disappear may be cut off (fragmented) from the main cluster. One can measure the size of the largest cluster as a fraction of the total system size, when a fraction of the nodes are removed from PINs [10]. The simulations we have performed can be extended to this topological measure too. This issue is under investigation currently.

Acknowledgements

One of the authors, Ka-Lok Ng, would like to thank the National Science Council to support this work. This work is supported by the R.O.C. National Science Council grants NSC 94-2745-E-468-008-URD.

References

1. Jeong H., Tombor B., Albert R., Oltvai Z.N., and Barabasi A.L. (2000). The large-scale organization of metabolic networks. *Nature* 407, 651.
2. Jeong H., Mason S., Barabási A.-L. and Oltvai Zoltan N. (2001). Lethality and central-ity in protein networks. *Nature* 411, 41.
3. Farkas I., Jeong H., Viscek T., Barabasi A.L and Oltvai Z.N., (2003). The topology of the transcriptional regulatory network in the yeast, *S. cerevisiae*. *Physica A*, 318, 601.
4. Albert R., Jeong H., and Barabasi A.L. (1999). Diameter of the World Wide Web. *Nature* 401, 130.
5. Kumar R., Raghavan P., Rajalopagan S., and Tomkins A., (2000). Proceedings of the 9th ACM Symposium on Principles of Databases Systems, 1 (Association for Computing Machinery, New York).
6. Faloutsos M., Faloutsos P., and Faloutsos C., (1999). On power-law relationships of the internet topology. *Proc. ACM SIGCOMM, Comput. Commun. Rev.* 29, 251.
7. Watts D.J., and Strogatz S.H. (1998). Collective dynamics of 'small-world' networks. *Nature* 393, 440.
8. Wasserman, S. and Faust, K. (1994). *Social Network Analysis*. Cambridge Univ. Press, Cambridge.
9. Barkai N and S. Leibler (1997). Robustness in simple biochemical networks. *Nature* 387, 913.
10. Albert R. Jeong H., and Barabasi A.L. (2000). Error and attack tolerance of complex networks. *Nature* 406, 378.
11. Sole RV, Montoya JM. (2001). Complexity and fragility in ecological networks. *Proc R Soc Lond B Biol Sci.* 268, 2039.
12. Xenarios I., Frenandez E., Salwinski L., Duan X., Thompson M., Marcotte E., and Eisenberg D., (2001). DIP: The Database of Interacting Proteins. *Nucl. Acid Res.* 29, 239.
13. Erdos P and Renyi A. (1960). On the Evolution of Random Graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5, 17.
14. Albert R. and Barabasi A.L. (2002). Statistical Mechanics of Complex Network. *Rev. Mod. Phys.* 74, 47.
15. Lee Po-Han, Huang Chien-Hung, Fang Jywe-Fei, Liu Hsiang-Chuan, Ng Ka-Lok (2005). Hierarchical and Topological Study of the Protein-protein Interaction Networks. (to appear at *Advances in Complex Systems*).
16. Coates P.J., and Hall P.A. (2003). The yeast two-hybrid system for identifying protein-protein interactions. *J Pathol.* 199(1), 4.
17. Lodish H., Berk A., Zipursky S., Matsudaira P., Baltimore D. and Darnell J. (2001). *Molecular Cell Biology*, 4th ed., W.H. Freeman, N.Y..
18. Maslov S, and Sneppen K. (2002). Protein interaction networks beyond artifacts. *FEBS Lett.* 530, 255.

19. Maslov S, and Sneppen K. (2002). Specificity and Stability in Topology of Protein Networks. *Science*, 296, 910.
20. Sneppen K., Maslov S., and Eriksen K.A. (2003). Analyzing Molecular Networks. Proc. Idea-Finding Symposium, Frankfurt Institute for Advanced Studies.
21. Shen-Orr S. S., Milo R., Mangan S. and Alon U. (2002). Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics* 31, 64.
22. Aloy P. and Russell R.B. (2002). FEBS Lett. Potential artifacts in protein-interaction networks. 530, 253.
23. von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* 417, 399.
24. Floyd R. W. *Communication of the ACM* 5, no.6, 345 (1962)

A Bayesian Approach for Integrating Transcription Regulation and Gene Expression: Application to *Saccharomyces Cerevisiae* Cell Cycle Data

Sudhakar Jonnalagadda and Rajagopalan Srinivasan*

Department of Chemical and Biomolecular Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
sudhakar@nus.edu.sg, chergs@nus.edu.sg

Abstract. The advent of high-throughput techniques is transforming biology into a data rich field. A variety of genomics data is now available, each providing a different perspective of gene regulation. Even though each type of data requires specific computational methods, methods that combine complimentary datasets are necessary to obtain additional information that is not available by analyzing the either of the dataset alone. In this paper, we propose a Bayesian approach to integrate gene expression data with genome-wide protein-DNA interaction data. The proposed method combines these datasets in order to probabilistic predict transcription factors for genes. We evaluate the proposed method using *Saccharomyces Cerevisiae* Cell Cycle data. Results are compared with that of previous method.

1 Introduction

Cells carryout their complex functions by timely altering the transcription rates of specific genes throughout the genome. The transcription rate of a gene is precisely regulated by the combinatorial action of several activator and repressor proteins called transcription factors (TFs) that bind to the promoter regions of genes and affect the binding of RNA polymerase [1]. The primary goal of biological studies is to understand gene regulation and to identify which transcription factors regulate which genes. Such insights are essential to develop models that predict cell responses to novel conditions. Even though analysis of genome-wide expression profiles enhances our understanding of cellular processes, there are certain inherent challenges when assigning regulators for genes. Microarray expression profiling does not distinguish between effect of direct binding of TF to a target gene and the indirect effect caused by intermediate TFs. So genes can have similar expression profile even though their regulators are different. Hence clustering of co-expressed genes [2, 3 and 4] is of limited use [5]. Segal et al. [6] proposed more advanced method to identify the targets of regulators using expression data. Their approach assumes that expression profile of regulated genes depends on expression of their regulators. This assumption is not always valid, for example during post-transcriptional modifications of TFs where the

* Corresponding author.

expression of regulator does not change appropriately. Hence expression data alone is not adequate for identifying the regulators for genes.

However, there are other genomic data sources that provide complementary information about TF-gene interactions. For example, the genome-wide location analysis method [7] identifies the direct protein-DNA physical interactions at genome-scale by combining the chromatin immunoprecipitation (ChIP) procedure with microarrays. Though location data is highly useful, false positives and false negatives hinder the assignment of TFs to genes. For instance, there is only moderate agreement between the genome-wide location studies of *Saccharomyces Cerevisiae* by Iyer et al. [8] and Simon et al. [9] for the same TFs: *mbp1*, *swi4*, and *swi6* [10]. However, by integrating gene expression and genome-wide location data one can extract useful and reliable information about regulation of genes.

There are two reported approaches to combine these two datasets. The first approach, proposed by Hartemink et al. [11], uses Bayesian networks with the location data influencing the model prior and the expression data influencing the likelihood. The identified network provides the links between TFs and their target genes. As another approach, Bar Joseph et al. [5] proposed a method that compliments the expression data with location data to overcome the false negatives in location data. In their approach, location data is used to classify genes into different sets such that genes in each set are bound by the same TFs. Then for each such group, a minimum radius sphere (capturing the genes within the set) is found in gene expression data. Then the genes without any regulators (false negatives) in location data are classified into these sets if they fall in the sphere and have the combined probability of regulatory interactions lesser than the predefined threshold. One of the limitations of their method is the computational complexity of finding the minimum radius sphere in the high dimensional expression data. Furthermore, this method is not extendable to other datasets such as gene / promoter sequences. Here we propose a Bayesian approach that reliably assigns TFs to genes by combining genome-wide location data with gene expression. Our approach is based on statistical theory and can be directly extended to new types of data.

In Section 2, we describe the proposed method. We evaluate the proposed method using the *Saccharomyces Cerevisiae* Cell Cycle data. Results are shown in Section 3. Discussion and Conclusions are given in Section 4 and 5, respectively.

2 Methodology

The proposed method uses the genome-wide location data and gene expression data in an incremental way to reliably assign regulators to genes. The method is schematically shown in Figure 1. A model is first developed using genes for which high-confidence transcription factors are available in the location data. This model is then used for assigning TFs to the remaining genes (i.e. those without reliable transcription factor information) using expression similarity. There are three steps in the method: (1) Conversion of location data into binary values (2) Model development for genes with TFs in location data (3) Bayesian classification of the remaining genes using the model identified in step 2. We describe these three steps in the following sections.

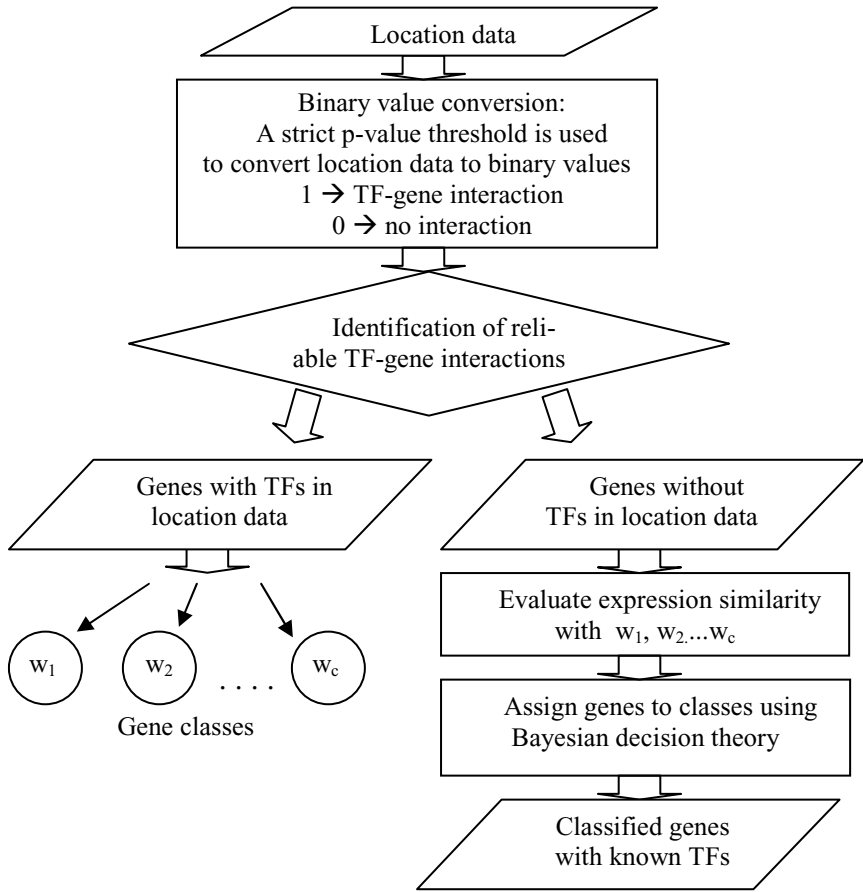


Fig. 1. The proposed methodology for integration of gene expression and genome-wide location data. Genes are first classified into several classes where each class of genes bound by same transcription factors (TFs). The unclassified genes are assigned to one of the existing classes using Bayesian decision rule.

2.1 Conversion of Location Data to Binary Values

The genome-wide location data contains the p-values for the TF-gene interactions. The lower the p-value, the higher the probability of interaction. These p-values have to be converted to binary values to decide where a particular TF binds to the gene or not. The value 1 indicates an interaction between a transcription factor and gene whereas the value 0 indicates no interaction. Binary conversion is carried out by selecting suitable threshold for p-value. Let $B_{m \times t}$ be the location data for m genes on t transcription factors where each element, b_{ij} , is the p-value for the interaction between gene i^{th} gene and j^{th} transcription factor. Then we consider the interaction between i^{th} gene and j^{th} transcription factor if b_{ij} is smaller than the p-value threshold P_T i.e.

$$b_{ij} = \begin{cases} 1 & \text{if } b_{ij} < P_T \\ 0 & \text{otherwise} \end{cases}. \quad (1)$$

2.2 Model Development for Genes with TFs in Location Data

A model consisting of TFs linked to expressions of the gene they regulate is developed in the next step. For this, by using a strict P_T , we can identify the most reliable interactions after binary conversion. Then different classes in the location data are identified such that all the genes within a class are bound by same TFs set. For this, the method searches for all the possible combinations of transcription factors (i.e. 2^t-1 combinations are possible with t TFs). For each such TFs set, our method finds the genes bound by all the TFs and considers them as a model component. Genes are allowed to be present in multiple model components. For example, a gene bound by regulators A,B and C in location data is allowed to be present in both the classes regulated by $\{A, B\}$, and $\{C\}$, respectively. However, subsets of a TFs set are not considered. In the above example, TF sets $\{A\}$ and $\{B\}$ would not be considered as possible TFs for that gene. The final model consists of all the model components thus identified.

2.3 Model-Based Bayesian Classification

After identification of reliable interactions and classification of genes, putative genes are assigned to the remaining genes using the above model using Bayesian classification rule. In general, Bayesian rule updates our belief of a hypothesis in the light of new evidence. In the present context, Bayesian rule updates the *a priori* probability (belief) that a previously unclassified gene belongs to the one of the classes (hypothesis) to a *posterior* probability using the expression similarity of the gene to the already classified genes (evidence).

Let $X_{m \times n}$ is the expression data matrix containing m genes measured on n time points. Assume that these m genes are classified into $w_i (1 \leq i \leq C)$ classes where all the genes in class w_i are bound by same set of transcription factors. Given a new gene with expression profile represented by x , the probability that x belongs to i^{th} class is given by Bayesian rule as [12]

$$P(w_i / x) = \frac{P(w_i) \cdot p(x/w_i)}{p(x)}. \quad (2)$$

where $P(w_i / x)$ is the *a posterior* probability of x belongs to class w_i , $P(w_i)$ is the *a priori* probability that x belongs to class w_i , $p(x/w_i)$ is the probability density function of x given the class w_i , and $p(x)$ is the probability density function of x given by

$$p(x) = \sum_{i=1}^C p(x/w_i) \cdot P(w_i). \quad (3)$$

The Bayes rule (Eq 2) shows how the measuring the expression profile of a gene changes the *a priori* probability to a *posterior* probability.

According to Bayesian theory, to reduce the probability error, a gene should assign to the class for which it has highest posterior probability i.e. assign x to class w_j if

$$p(w_j / x) > p(w_i / x) \quad \forall i \neq j. \quad (4)$$

The denominator in Eq. 2 is a normalization factor which makes the sum of posterior probabilities equals to 1. For classification purposes, it is not necessary to have the normalized posterior probabilities; hence the denominator is normally discarded from the analysis. Then the classification rule in Eq.4 becomes

$$p(x/w_j) \cdot P(w_j) > p(x/w_i) \cdot P(w_i) \quad \forall i \neq j. \quad (5)$$

In practice we can use any monotonous function of $p(x/w_i)P(w_i)$ that is convenient. In this case we used the logarithm of $p(x/w_i)P(w_i)$ represented by $g_i(x)$. The conditional probability function of x for a given class w_i , $p(x/w_i)$, is assumed to be multivariate normal distribution. Hence the Bayesian decision rule is given by

$$g_j(x) > g_i(x) \quad \forall i \neq j. \quad (6)$$

$$g_i(x) = -\frac{1}{2}(x - \mu_i)\Sigma_i^{-1}(x - \mu_i) - \frac{1}{2}\log(|\Sigma_i|) + \log P(w_i). \quad (7)$$

Where μ_i and Σ_i are the mean vector and covariance matrix of class w_i , respectively. $P(w_i)$ is the fraction of genes in class w_i . The mean and covariance matrix of a class are estimated using the samples in that class as

$$\mu_i = \frac{1}{n_i} \sum_{x \in w_i} x. \quad (8)$$

$$\Sigma_i = \frac{1}{n_i} \sum_{x \in w_i} (x - \mu_i)(x - \mu_i)^T. \quad (9)$$

where n_i is the number of samples in class w_i .

In Eq. 7, Bayesian rule is used with the Mahalanobis distance between the expression profile of gene x and the mean of the class μ_i along with the *a priori* probability and the class covariance matrix to generate the *a posterior* probability that a gene belongs to a class.

As a strict P_T is used in binary conversion of the location data, the gene interactions are identified with high confidence (few false positives). False negatives may be induced by the strict threshold; the proposed method reduces such false negatives by complimenting with gene expression data. For each gene with no regulators in location data, the proposed method uses its expression similarity to the already classified genes as evidence and generates the probability that it belongs to these classes.

Finally the gene is assigned to the class (set of TFs) for which it has highest similarity. Hence the proposed method reliably assigns the TFs to genes.

3 Results

We evaluate the proposed Bayesian approach to identify the regulators for *Saccharomyces Cerevisiae* cell-cycle regulated genes reported by Spellman et al. [13]. Spellman et al. measured the expression levels of Yeast genes at 73 time points during three independent conditions: α factor arrest, elutriation, and arrest of *cdc15*. They identified approximately 800 cell-cycle regulated genes using periodicity and correlation algorithms. The genome-wide location data for these genes are collected from Simon et al. [9]. Simon et al. conducted the genome-wide location study for nine known cell-cycle transcription factors: Fkh1, Fkh2, Ndd1, Mcm1, Ace2, Swi5, Mbp1, Swi4, and Swi6. Out of the 800 cell-cycle regulated genes location data is available for 794 genes. We use these 794 genes in this study.

We used the strict p-value threshold, P_T , of 0.001 to convert the p-values to binary values [5]. This means there is 0.1% probability that an interaction is happened by chance. Considering the false positives even at this strict threshold, we considered the only the classes containing at least 5 genes. Then we tested all the combination of the nine cell-cycle regulators for eligible gene classes. This procedure identified 28 classes containing 206 unique genes (out of these 794). The first three columns of Table 1 show the classes, class sizes and their regulators. Different classes contain different number of genes while the minimum size is 5 and maximum is 34.

In the third step, we calculated the probabilities for each of the remaining 588 genes belonging to all the 28 classes using Bayesian rule. The proposed approach needs the inverse of the covariance matrix of each class to generate the posterior probabilities (Eq.7). Since the dimensionality of the expression data is 73 (time points), we need at least 73 genes in each class to have the non-singular covariance matrix and hence the inverse. To solve this problem we used Principal Component Analysis (PCA). PCA is a multivariate technique that finds the principal components (directions) of variability in the data, and transforms the related variables into a set of uncorrelated ones. These principal components (PCs) are the linear combinations of original variables [14]. The first few PCs capture most of the variance in the data whereas the remaining PCs represent noise. Hence the dimensionality of the data can be reduced by considering the first a few PCs. We applied the PCA on the whole data before identifying the classes. Since the minimum size of the classes is 5, we used the first four PCs in order to have the non singular covariance matrix for all classes.

Then the 588 genes are assigned to one of these 28 classes using their highest posterior probabilities (Table 1, last column). The number of gene classified to different classes varies from zero to the maximum of 98. The distribution of normalized maximum *a posterior* probability of 588 genes is shown in Figure 2. 169 (out of 588) genes have the highest posterior probability of at least 0.5.

Table 1. Prediction of class labels for genes without any transcription factors in genome-wide location data. Genes are assigned to the class with highest posterior probability.

Class No.	Size	Transcription Factors (TFs)	No. of genes classified
1	5	Fkh2 Ndd1 Mcm1 Mbp1 Swi4 Swi6	0
2	6	Fkh2 Ndd1 Mbp1 Swi6	5
3	7	Fkh2 Ndd1 Swi4 Swi6	1
4	5	Fkh2 Mcm1 Swi4 Swi6	0
5	7	Fkh2 Ace2 Swi4 Swi6	1
6	12	Fkh2 Mbp1 Swi4 Swi6	14
7	5	Mcm1 Mbp1 Swi4 Swi6	9
8	5	Ace2 Swi5 Mbp1 Swi6	4
9	6	Ace2 Swi5 Swi4 Swi6	5
10	5	Ace2 Mbp1 Swi4 Swi6	7
11	5	Swi5 Mbp1 Swi4 Swi6	1
12	9	Fkh2 Ndd1 Mcm1	18
13	6	Fkh1 Fkh2	2
14	6	Fkh2 Ace2	3
15	6	Fkh2 Swi5	18
16	5	Fkh2 Swi4	19
17	7	Fkh2 Swi6	1
18	9	Ace2 Swi5	18
19	8	Mbp1 Swi4	19
20	13	Mbp1 Swi6	50
21	28	Swi4 Swi6	50
22	16	Fkh1	98
23	16	Ndd1	65
24	22	Mcm1	76
25	34	Swi5	35
26	7	Mbp1	30
27	16	Swi4	31
28	5	Swi6	8

Here we give a brief analysis of the classification results. Spellman et al. clustered cell-cycle regulated genes into different clusters based on their similarity in expression over all experiments. We analyze the results for some of these clusters:

CLN2 Cluster: The CLN2 cluster contains 76 genes that show peak expression during mid-G₁ phase in their expression. These genes are regulated by MBF (complex of Mbp1 and Swi6) and SBF (complex of Swi4 and Swi6) [13]. TFs are available for 29 (out of 76) genes in location data, but no regulators are found for remaining 47 genes. The proposed method correctly identifies the regulators for these genes. Our approach

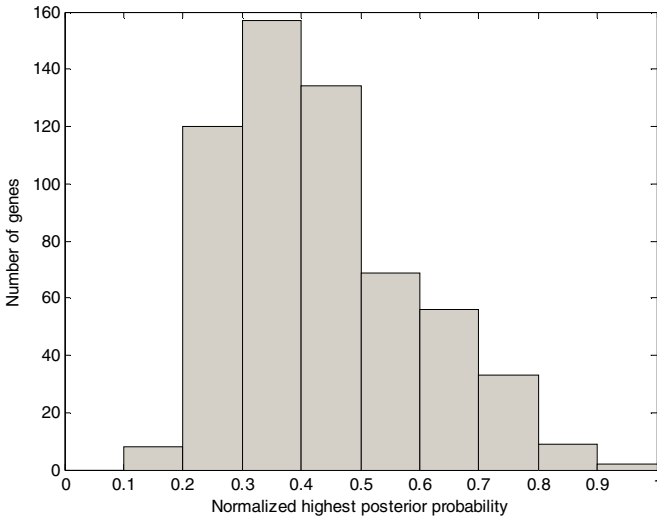


Fig. 2. The distribution of normalized maximum a posterior probability of 588 genes whose regulators are predicted using the proposed method

assigns either MBF or SBF or both as the regulators for 37 of these genes. These genes include POL12, POL30, CDC9, and STB1 etc. For the remaining genes, one of the subunits of SBF and MBF are assigned.

CLB2 cluster: The CLB2 cluster contains 36 genes regulated by the complex formed the transcription factors Mcm1, Ndd1, Fkh1/Fkh2 [15, 16]. Regulators are not available for 15 of these genes in genome-wide location data. Our approach identifies all three TFs Mcm1, Ndd1, and Fkh1/Fkh2 as the regulators for 12 out of these 15 genes. These genes include CLB1, MOB1, and HOF1 etc. Ndd1 is assigned as a regulator for 2 genes and Fkh1 is assigned for the remaining one gene.

MCM cluster: The MCM cluster contains 34 genes regulated by Mcm1 [13]. Our approach predicted the transcription factors for 23 out of these 34 genes. Comparing to the other clusters, the results for this clusters are not accurate. Mcm1 is assigned as a transcription factor for 9 genes and Ndd1 for 7 genes. One or more of the Fkh2, Ace2, Swi4, and Swi5 are assigned to the remaining genes.

The application of Bar-Joseph et al. procedure for these same datasets yielded 34 classes with a mean class size of around 9. Only 22 of the 76 genes from CLN2 cluster are included in these 34 classes. Similarly 19 and 15 genes from CLB2 and MCM clusters included in these 34 classes. Moreover, these 22, 19, and 15 genes are distributed over several classes giving no clear clue of regulators for these genes.

4 Discussion

In this paper, we proposed a Bayesian approach for combining genome-wide location data with gene expression data for identifying regulators for genes. Application of proposed approach for *Saccharomyces Cerevisiae* Cell-Cycle data revealed its

efficacy. However, there are several issues to be addressed. The first one is the low sample situation. Out of these 794 genes used in this study, only 206 genes have reliable TFs in location data whose expression data is later used to identify the parameters (mean and covariance matrices). The minimum size of class is 5. The estimation of the parameters generally needs more genes in each class to nullify the effect of noise in the data. This problem can be eliminated by using the same covariance matrix for all the classes. Then the parameters can be reliably estimated by pooling the genes from all the classes. This also eliminates the need for PCA. This needs further examination. Nevertheless, for the case considered in this paper, the proposed method showed reasonable performance.

In this paper all the genes with no regulators in location data are assigned to one of the predefined classes based on their *posterior* probability. Even though the results are reasonably correct, it is preferable to develop criterion to reject genes in case no significant evidence is available for classification. From Figure 2, it is evident that some of the genes have the maximum *posterior* probability less than 0.5 indicating that they do not have significant evidence to assign to any class. Hence, it is better not to assign these genes to any of the classes. This can be done by selecting a suitable threshold for maximum *a posterior* probability.

Also, the proposed method needs regulator information for some genes which is used to find TFs for other genes. In this paper we used genome-wide location data for this purpose. Such information can also be extracted from other sources, such as literature search and promoter sequence analysis. Since the proposed Bayesian approach uses some sort of new evidence to convert the *a priori* probabilities to *a posterior*, it is relatively easy to extend this method to other complimentary datasets. For example, if we know that a particular gene has a regulatory element similar to that of a set of other genes, we can use this as additional evidence (similar to expression profile similarity). The same procedure can be followed for other complimentary data. This work is in progress.

5 Summary and Conclusions

In this paper, we proposed a Bayesian approach to integrate genome-wide location data with gene expression data to predict the regulators for genes. The proposed method has been evaluated by predicting the regulators for *Saccharomyces Cerevisiae* Cell Cycle regulated genes. The proposed method showed reasonable performance and correctly predicted the regulators for several genes. Yet a comprehensive analysis of proposed method using gold standard datasets and comparison with other methods is necessary to establish the method. The extension of the proposed method to integrate other complimentary data is also to be tested.

References

1. Lee, T.I. and Young, R. A.: Transcription of eukaryotic protein-coding genes. *Annu. Rev. Genet.* (2000) 34:77-137
2. Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, (1998) 95:14863-14868

3. Tamayo,P., Slonim,D., Mesirov,j., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA.* (1999) 96:2907-2912.
4. Tavazoie,S., Huges,J.D., Campbell,M.J., Cho,R.J. and Church,G.M.: Systematic determination of genetic network architecture. *Nat. Genet.* (1999) 22:281-285.
5. Bar-Joseph,Z., Gerber,G. K., Lee,T. I., Rinaldi,N. J., Yoo,J. Y., Robert,F., Gordon,D.B., Fraenkel,E., Jaakkola,T. S., Young,R. A. and Gifford,D. K.: Computational discovery of gene modules and regulatory networks. *Nat. Biotechnol.* (2003) 21:1337-1342
6. Segal, E., Shapira, M., Regev, A., Peer, D., Bostein, D., Koller, D. And Friedman, N.: Module networks: identifying regulatory modules and their condition- specific regulators from gene expression data. *Nat. Genet.* (2003) 34:166-176
7. Ren,B., Robert,F., Wyrick,JJ., Aparicio,O., Jennings,EG., Simon,I., Zeitlinger,J., Schreiber,J., Hannett,N., Kanin,E., Volkert,TL., Wilson, C.J., Bell, S.P. and Young, R.A.: Genome-wide location and function of DNA binding proteins. *Science.* (2000) 290:2306-2309
8. Iyer, VR., Horak,C,E., Scafe, C,S., Botstein,D., Snyder,M., Brown,P.O.: Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* (2001) 409:533-538
9. 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. and Young,R.A.: Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell* (2001) 106:697-708
10. Futcher B.: Transcriptional regulatory networks and the yeast cell cycle. *Curr. Opin. Cell. Biol.* 2002, 14:676-683
11. Hartemink, A.J., Gifford, D.K., Jaakkola, T.S. and Young, R.A.: Combining location and expression data for principled discovery of genetic regulatory network models. In *Pacific Symposium on Biocomputing* (2002) 437-449
12. Duda, R.O, and Hart, M.P *Pattern classification and scene analysis.* New York (1973) Wiley.
13. Spellman,P.T., Sherlock., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell.* (1998) 9:3273-3297
14. Jackson, J. E.: *A User's Guide to Principal Components.* John Wiley (1991) NY.
15. Koranda,M., Schleiffer,A., Endler,L. and Ammerer,G.: Fork-head-like transcription factors recruit Ndd1 to the chromatin of G2/M-specific promoters. *Nature* (2000). 406:94-98
16. Zhu,G., Spellman,P.T., Volpe,T., Brown,P.O., Botstein,D., Davis,T.N., and Futcher, B.: Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature* (2000) 406: 90-94

Probabilistic *in Silico* Prediction of Protein-Peptide Interactions

Wolfgang Lehrach^{1,2}, Dirk Husmeier¹, and Christopher K.I. Williams²

¹ Biomathematics and Statistic Scotland, UK

² Institute of Adaptive and Neural Computation, University of Edinburgh, UK

Abstract. Peptide recognition modules (PRMs) are specialised compact protein domains that mediate many important protein-protein interactions. They are responsible for the assembly of critical macromolecular complexes and biochemical pathways [Pawson and Scott, 1997], and they have been implicated in carcinogenesis and various other human diseases [Sudol and Hunter, 2000]. PRMs recognise and bind to peptide ligands that contain a specific structural motif. This paper introduces a novel discriminative model which models these PRMs and allows prediction of their behaviour, which we compare with a recently proposed generative model. We find that on a yeast two-hybrid dataset, the generative model performs better when background sequences are included, while our discriminative model performs better when the evaluation is focused on discriminating between the SH3 domains. Our model is also evaluated on a phage display dataset, where it consistently out-performed the generative model.

1 Introduction

Peptide recognition modules (PRMs) are specialised compact protein domains that mediate many important protein-protein interactions. They are responsible for the assembly of critical macromolecular complexes and biochemical pathways [Pawson and Scott, 1997], and they have been implicated in carcinogenesis and various other human diseases [Sudol and Hunter, 2000]. PRMs recognise and bind to peptide ligands that contain a specific structural motif. One of the most actively studied PRMs is the SH3 domain, which binds to peptide ligands that contain a particular proline-rich core. [Tong et al. 2002] carried out two extensive experimental studies to infer the network of SH3-mediated protein-protein interactions in *Saccharomyces cerevisiae*. They identified 28 SH3 domain proteins in the *S. cerevisiae* proteome, which were used as baits and screened against conventional and Proline-rich libraries in a yeast two-hybrid (Y2H) experiment. In a second independent study, they screened random peptide libraries by phage display to identify the consensus sequence for preferred ligands that bind to each PRM (see [Twyman 2004] for a description of Y2H and phage display methods). Based on these consensus sequences, they inferred a protein-protein interaction network that links each PRM to proteins containing the preferred ligand. Since both experimental procedures are intrinsically noisy, the two

independently inferred interaction networks were found to show only a modest degree of overlap.

Reiss and Schwikowski [2004] addressed the question of whether computational *in silico* approaches would allow some of the difficult and expensive experimental procedures to be more specifically targeted, or even bypassed altogether. To this end, they developed a probabilistic generative model of the SH3 ligand peptides, based on the widely used Gibbs sampling motif finding algorithm [Lawrence et al., 1993, Liu et al., 1995]. Reiss and Schwikowski [2004] encouragingly demonstrate that a probabilistic model trained on protein sequences and observed physical interactions can succeed in independently predicting new protein-protein interactions mediated by SH3 domains. However, a shortcoming of their model is a dependence on tuning parameters that have to be chosen in advance by the user and that are not inferred from the data. Inappropriate values reduce the performance of their algorithm to using standard motif searching algorithms, and it is unlikely that universal values applicable to different protein (super-) families exist. Also, the proposed model borrows substantial strength from a heuristic discriminative modification of the prior, which again depends on various tuning parameters.

This paper proposes an alternative *in silico* method for the prediction of SH3-mediated protein-protein interactions, which addresses some of the shortcomings of the model introduced by Reiss and Schwikowski [2004]. A key feature of our model is its focus on distinguishing between different SH3 binding domains and our use of a discriminative model to distinguish the common SH3 binding domain from the background. This is in contrast to the approach of Reiss and Schwikowski [2004], which is based on a generative model of the whole sequence. As discussed in Segal and Sharan [2004], a generative approach can be confounded by repetitive or over-represented motifs that are unrelated to PRM-peptide interactions, which our discriminative model avoids by formulating the learning problem in terms of a supervised classification problem.

The model we propose is inspired by a DNA-sequence model developed by Segal et al. [2002] and Segal and Sharan [2004]. However, our model incorporates additional prior information by assuming that there is an underlying generic motif to which all specific SH3 domains are related. This splits the problem into separate stages of discrimination against background and discrimination between the SH3 domain stages, where the discrimination against background has comparatively low computational costs and is followed by explicitly modelling the differences between the binding sites for each SH3 domain.

Additionally, due to the larger size of the alphabet (20 amino acids instead of 4 nucleotides) and the small number of interactions per SH3 domain, the maximum likelihood approach to parameter estimation is susceptible to overfitting. An important component of our approach, therefore, is the inclusion of a regularisation scheme, resulting in a maximum a posteriori (MAP) approach. Additionally, we train an ensemble of models, which also reduces overfitting.

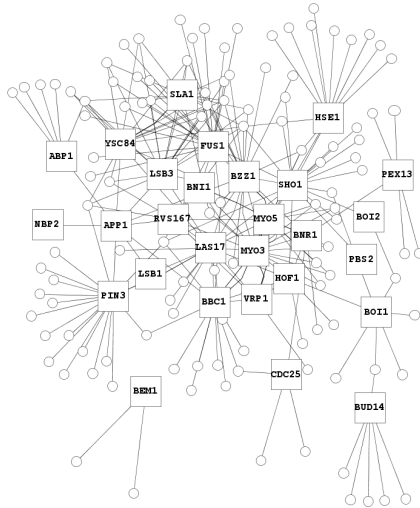


Fig. 1. The yeast two hybrid interaction network of SH3. The labelled squares represent the central SH3 domains, while the circles represent the peripheral proteins that were found to bind to the SH3 domains. We have left out the labels on the peripheral proteins for lack of space. A full size version is available from <http://lehrach.com/wolfgang/recomb06satellite>, as well as a similar network obtained from the phage display experiments.

2 Methods

In this section we define the problem, show how to derive our discriminative model and then describe how to apply it. Let $\mathbf{D} = \{d_i\}$ denote a set of SH3 domains, and $\mathbf{S} = \{s_j\}$ a set of protein sequences. We introduce a binary variable $\epsilon_{ij} \in \{0, 1\}$, where $\epsilon_{ij} = 1$ indicates that sequence s_j binds to SH3 domain d_i , while $\epsilon_{ij} = 0$ indicates the absence of an interaction. We assume that we are given a protein interaction network $\mathbf{E} = \{\epsilon_{ij}, d_i \in \mathbf{D}, s_j \in \mathbf{S}\}$ from a Y2H or phage display experiment where s_j refers to the j th protein sequence and $s_{j,k}$ refers to the amino acid in the k th position of the j th sequence. The objective is to derive a model that predicts this network from the sequences alone.

The central assumption of the model is that all of the binding sites of the SH3 domain are sufficiently similar to allow a single generic motif detector to find the binding sites for all SH3 domains, and furthermore that each binding site only binds to a single SH3 domain. These assumptions are expressed as $P(\epsilon_{i,j} = 1|s) = P(M_j = 1, O_j = i|s)$, where M_j indicates if the generic motif that represents the binding site is present in the j th sequence and O_j represents which SH3 domain the binding site interacts with.

The unknown position of the motif on the j th sequence is represented by the variables a_j (marginalised over later). It holds that:

$$P(O_j = i, M_j = 1, a_{i,j} = k | s_j) = P(O_j = i | M_j = 1, a_j = k, s_j) \times P(M_j = 1, a_j = k | s_j), \quad (1)$$

where both terms are derived from a generative model. The second term models the probability of the generic motif appearing in the k th position. The probability of the part of sequence containing the motif is defined as:

$$P(s_{j,k}, s_{j,k+1}, \dots, s_{j,k+p} | M_j = 1, a_j = k) = \prod_{m=1}^p \phi_{m, s_{k+m}}, \quad (2)$$

where $\phi_{m,c}$ is the probability of the generic binding motif containing in the m th position the c th amino acid and p is the length of the motif. The probability of the c th amino acid occurring in any part of the sequence which is not in a motif is φ_c , which means the probability of a sequence without the generic binding site is $P(s_j | M_j = 0) = \prod_{q=1}^{n_j} \varphi_{s_{j,q}}$ where n_j is the length of the j th sequence. The probability of the whole sequence given that it contains the generic binding site is then $P(s_j | M_j = 1, a_j = k) = B_k \prod_{q=1}^{n_j} \varphi_{s_{j,q}}$ where $B_k = \prod_{m=1}^p \frac{\phi_{m, s_{j,k+m}}}{\varphi_{s_{j,k+m}}}$. B_k is the likelihood ratio of the k th possible motif position being part of the motif as opposed to the background. The second term from Equation (II) is:

$$P(M_j = 1, a_j = k | s_j) = \frac{P(s_j | M_j = 1, a_j = k) P(M_j = 1) P(a_j = k)}{\sum_{d,e} P(s_j | M_j = d, a_j = e) P(M_j = d) P(a_j = e)} = \frac{\frac{1}{n_j - p + 1} B_k}{\frac{1}{n_j - p + 1} \sum_{e=1}^{n_j - p + 1} B_e + \frac{P(M_j=0)}{P(M_j=1)}}, \quad (3)$$

where $P(M_j, a_j) = P(M_j) P(a_j)$ and we assume a uniform binding motif position prior $P(a_j = k) = \frac{1}{n_j - p + 1}$.

The first term in Equation (II) is the probability of a binding site binding to a specific SH3 domain. As before, we first generatively model the sequences piece that contains the i th SH3 domain binding site:

$$P(s_{j,k}, s_{j,k+1}, \dots, s_{j,k+p} | O_j = i, a_j = k) = \prod_{m=1}^p \theta_{i,m, s_{k+m}}, \quad (4)$$

where $\theta_{i,m,c}$ is the probability of the binding site of the binding site motif of the i th SH3 domain containing in the m th position the c th amino acid. The probability of the whole sequence is then $P(s_j | O_j = i, a_j = k) = \prod_{q=1}^{n_j} \varphi_{s_{j,q}} \prod_{m=1}^p \frac{\theta_{i,m, s_{j,k+m}}}{\varphi_{s_{j,k+m}}}$. Applying Bayes' rule removes the dependence on the background and gives:

$$P(O_j = i | s_j, a_j = k) \propto R_{i,k}, \quad (5)$$

where $R_{i,k} = P(O_j = i) \prod_{m=1}^p \theta_{i,m, s_{k+m}}$. Combining Equations (5), (II) and (3), and marginalising over the unknown motif position a_j gives:

$$P(O_j = i, M_j = 1 | s) = \frac{\frac{1}{n_j - p + 1} \sum_{k=1}^{n_j - p + 1} \frac{R_{i,k}}{\sum_j R_{j,k}} B_k}{\exp\{-T\} + \frac{1}{n_j - p + 1} \sum_{k=1}^{n_j - p + 1} B_k}. \quad (6)$$

We now define the weights and the threshold for detecting the generic SH3 domain binding site as $W_{m,.} = \log \frac{\theta_{m,.}}{\varphi}$ and $T = \log \frac{P(M_j=1)}{P(M_j=0)}$. The corresponding weights and thresholds for discrimination between the binding sites are defined as $W_{i,m,.} = \log \theta_{i,m,.}$ and $T_i \propto \log P(O_j = i)$.

2.1 Parameter Estimation

Having specified the model, we next need to estimate the parameters, which are the set of weights $\mathbf{W} = \{W_{k,l}, W_{i,k,l}\}$ and thresholds $\mathbf{T} = \{T, T_i\}$. A standard way to optimise these parameters, adopted for instance in Segal et al. [2002] and Segal and Sharan [2004], is to follow a maximum likelihood approach. Given the training data \mathbf{E} , we want to maximise the log likelihood:

$$\log P(\mathbf{E}|\mathbf{W}, \mathbf{T}) = \sum_{i,j} \epsilon_{i,j} \log P(\epsilon_{i,j} = 1 | s_j, \mathbf{W}, \mathbf{T}) + (1 - \epsilon_{i,j}) \log (1 - P(\epsilon_{i,j} = 1 | s_j, \mathbf{W}, \mathbf{T})). \quad (7)$$

It is straightforward to derive the partial derivatives for Equation (7). This allows us to apply an iterative gradient-based optimisation scheme, the details of which are omitted for lack of space.

2.2 Regularisation

A shortcoming of the maximum likelihood approach discussed in the previous section is its susceptibility to over-fitting. This might not be so much of a problem when applied to DNA Segal and Sharan [2004], Segal et al. [2002], but it becomes a serious issue for proteins due to the extended alphabet of 20 rather than 4 letters. A standard approach widely applied in machine learning is to impose a prior probability on the weights \mathbf{W} such that large weight values are discouraged and an *a priori* value of zero is assumed. A weight being set to 0 corresponds to a specific amino acid in a given position not being informative as to whether or not that position is part of the binding motif. Considering the paucity of the data, this is likely to be a common occurrence¹. We use the Laplacian prior from Williams [1995]:

$$P(\mathbf{W}|\alpha) \propto \exp \left(-\alpha \sum_{i,k,l} |W_{i,k,l}| \right) \quad (8)$$

The principal advantage of the Laplacian compared to the standard Gaussian regularisation, where $P(\mathbf{W}) \propto \exp \left(-\frac{1}{2}\alpha \sum_{i,k,l} W_{i,k,l}^2 \right)$, is that the Laplacian penalises large weights less severely, while small weights are more strongly driven down to 0. This corresponds to our prior belief that the binding site motifs will

¹ An alternative view is that we wish to find the simplest explanation for what is needed to detect the binding site.

strongly depend on certain amino acids occurring in certain positions whereas most other positions are not involved in binding.

Instead of maximising the likelihood [Equation (7)], we maximise the posterior probability $P(\mathbf{W}, \mathbf{T}|D) \propto P(D|\mathbf{W}, \mathbf{T}) P(\mathbf{W}) P(\mathbf{T})$. The prior probability of the weights is obtained from Equation (8) by integrating out the hyperparameter, $P(\mathbf{W}) = \int P(\mathbf{W}|\alpha) P(\alpha) d\alpha$, as discussed in Williams [1995]. The thresholds \mathbf{T} are unregularised corresponding to a uniform distribution; see Williams [1995].

The complete optimisation procedure is repeated from ten random initialisations. The resulting models are sorted by their posterior probability and the top half are used to create an ensemble. A prediction for an edge is then the mean prediction from the models in the ensemble.

3 Simulations

We evaluated the performance of the proposed discriminative model and the existing generative model of Reiss and Schwikowski [2004] on the phage display and Y2H protein interaction data of Tong et al. [2002]. The generalisation performance was evaluated using 10-fold cross-validation.

For comparison with Reiss and Schwikowski [2004], we measured the performance in terms of ROC curves, which are obtained by subjecting the predicted posterior probabilities $P(\epsilon_{ij}|s_j)$ to various threshold parameters $\theta \in [0, 1]$. By numerically integrating over the whole parameter range $\theta \in [0, 1]$ we obtain the area under the ROC curve (AUROC).

The ability of the generative model to learn the differences between the SH3 binding domains is poorly reflected in the evaluation of Reiss and Schwikowski [2004], where all of the sequences not containing the SH3 domain binding site are included in their evaluation. As an illustration, consider a model that correctly predicts every instance of an SH3 binding domain, but without distinguishing between different SH3 domains, which corresponds to what the authors refer to as the *global* model. In terms of predicting actual protein interaction networks, as depicted for example in Figure 1, the performance of this predictor is poor; it will either predict an interaction with none or with all SH3 domain proteins. However, due to the large number of non-interacting sequences its AUROC score will be close to the optimal value of 1.0. Consequently, large AUROC scores do not indicate a reliable prediction of the actual SH3 protein interaction network. To avoid this problem, we also perform an evaluation only on the binding partners of the SH3 domains; these are the proteins that correspond to the nodes shown in Figure 1.

When evaluating the classifier with background sequences (the same evaluation as Reiss and Schwikowski [2004]), it is of particular interest if the classifier can predict positive interactions with a low rate of false positives. In order to evaluate this, we additionally calculate the AUROC01 score, which is the proportion of the possible area filled under the ROC curve given that the rate of false positives is smaller than 0.1. When evaluating without background

sequences, the total number of non-interactors is considerably reduced, and the total AUROC becomes more informative.

We refer to the model of [Reiss and Schwikowski \[2004\]](#) as the generative model. This model depends on various tuning parameters, which are not inferred from the data but rather have to be set by the user in advance. For our comparative study, we used the default values defined in their software. These parameters were optimised by [Reiss and Schwikowski \[2004\]](#) for the yeast two-hybrid SH3 interaction network; hence they should reflect a quasi-ideal performance.

The models are compared against the generic discriminative model, which does not distinguish between the SH3 domains but only looks for motifs indicating that any SH3 domain binding site is present. This model is obtained by marginalising out O_j from Equation (6). When discriminating between SH3 domains, the naive classifier is also shown, which predicts each interaction as the prior frequency of that SH3 domain binding.

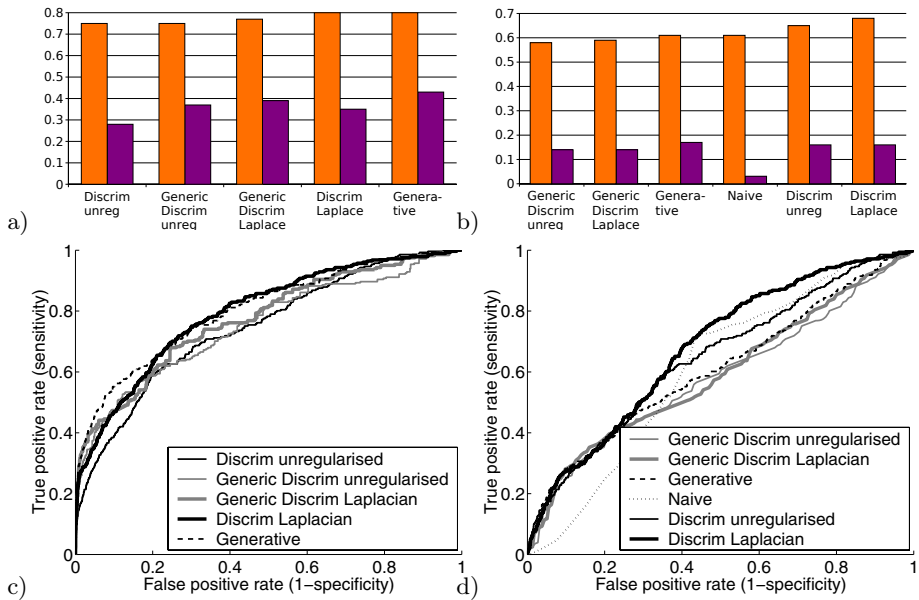


Fig. 2. A comparison of the performance of the discriminative and generative models on the yeast two-hybrid network. Sub-figure a) shows the performance of the various classifiers when all background sequences are included. The left bar for each model represents the AUROC score, while the right bar represents the AUROC01 score. Sub-figure c) shows the corresponding ROC curves. Sub-figure b) shows the performance of the models without background sequences, to focus the evaluation on the ability of the models to distinguish between the different SH3 domains. Sub-figure d) again shows the corresponding ROC curves. For space reasons, “Discriminative” has been shortened to “Discrim”. The generative model exhibits slightly superior performance when evaluating with background, while the discriminative model performs better when evaluating without background sequences. Regularisation improves performance, as does using the full model as opposed to only using the generic model.

4 Results

The performance of the models on the yeast two-hybrid network (see Figure 1) is outlined in Figure 2. When the non-binding sequences are included in the evaluation, the vast majority of possible interactions do not occur, implying that the AUROC01 score is more interesting as discussed in Section 3. Hence the generative model exhibits a slightly better performance due to its AUROC01 score. When non-binding sequences are excluded, the overall score becomes more important as there are far fewer non-interacting sequence pairs. Here, the discriminative model performs better due to its superior AUROC score.

The results indicate the importance of regularisation, as the performance of the unregularised models is inferior to the Laplacian regularised model. The performance increases slightly when the model is trained to distinguish between the SH3 domains, as can be seen by the inferior performance of the generic models.

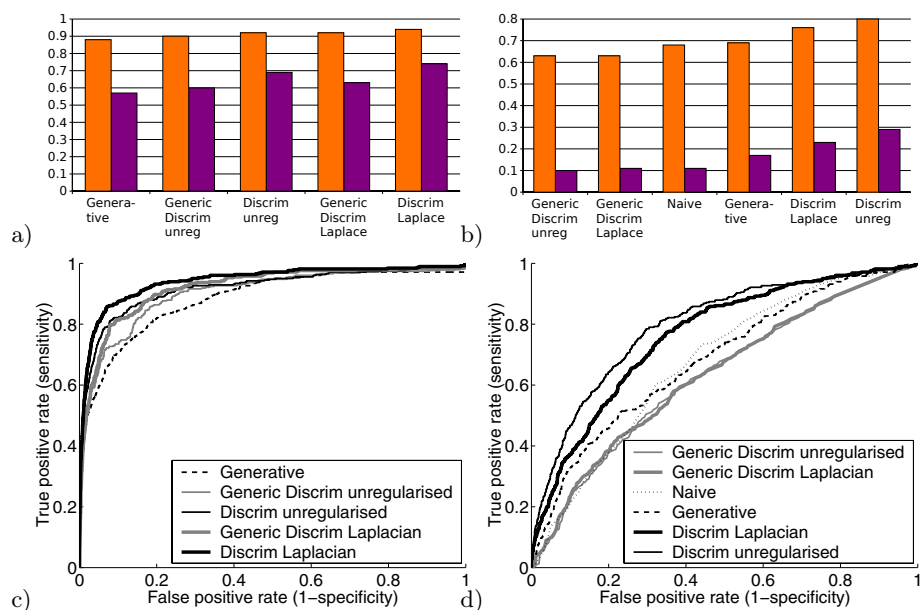


Fig. 3. Evaluating the performance of the models on the phage display dataset. As before, Sub-figure a) shows the performance of the various models when all background sequences are included, where the height of the left bar represents the AUROC score for each model and the height of the right bar represents the AUROC01 score. Sub-figure c) shows the ROC curves. Sub-figure b) evaluates the model without background sequences, and sub-figure d) shows the corresponding ROC curves, which focuses the evaluation on the ability of the models to distinguish between the different SH3 domains. Here the discriminative model performs significantly better than the model of [Reiss and Schwikowski \[2004\]](#), both with and without background sequences, measured by both the AUROC and AUROC01 score. Using the full model as opposed to the generic motif detector gives a relatively large increase in performance.

Figure 3 summarises the results obtained on the phage display dataset. As opposed to the yeast two-hybrid data, where the Laplacian-regularised model clearly outperformed the unregularised model, the effect of regularisation is less clear on the phage display data. While regularisation does improve the overall classification performance on the complete set of sequences (Figure 3, top left panel), we found that when discriminating between the different SH3 domains, regularisation actually turned out to be slightly counter-productive (Figure 3, top right panel). Note that this effect is a consequence of using an ensemble of models. It can be explained in terms of the bias-variance decomposition of the generalisation error, where an ensemble of individually overfitting predictors may substantially reduce the variance term; see Sollich and Krogh [1996] and Husmeier and Althoefer [1998] for further details.

The benefit from using the full model as opposed to only the generic binding site detector is more pronounced. This is probably due to the better match between the phage display data and the model, and the fact that yeast two-hybrid is known to be very noisy.

5 Discussion

The work presented in the present article has been motivated by Reiss and Schwikowski [2004], who developed a probabilistic sequence model based on the Gibbs motif sampler. Our alternative discriminative approach removes the need for hand-tuning heuristic parameters, allowing easy application to novel datasets. Both models perform sufficiently well in discriminating between binding and non-binding sequences, with large AUROC scores and large slopes of the ROC curves for low false positive values. The discriminative model performed better on the phage display data, while the generative model performed slightly better on the Y2H dataset.

The task of discriminating between the different SH3 domains is substantially harder, owing to two reasons: the training set is much smaller, and the binding motifs are quite similar (e.g. all exhibit a proline-rich core), requiring the model to pick up on subtle differences between them. The ROC curves obtained for the discriminative task (right panels in Figures 2 and 3) are noticeably better than what would have been obtained by chance. This is an encouraging finding that should stimulate future work on *in silico* prediction methods. On the discriminative task, the model proposed in this paper outperforms the generative model on both the yeast two-hybrid as well as the phage display data. Hence, it makes an important contribution towards the actual identification of the protein interaction network, as opposed to only discriminating between binding and non-binding sequences.

The model promises to provide biologically relevant information like predictions of locations of the binding sites. The generic motif detection weights highlight how to detect a binding site, while the other weights describe how to distinguish between the SH3 domains, also of biological interest. The discriminative basis of the model encourages focusing on distinguishing between the

binding motifs that mediate the different PRM-peptide interactions, allowing the model to pick up on faint but potentially important differences which could otherwise be lost with the heuristic and parameterised discrimination used by Reiss and Schwikowski [2004].

Note that our model can equally be applied to the prediction of protein-DNA interactions and the identification of transcription factor binding sites. In general, this discriminative model should be applicable to the modelling and recognition of many different regulatory elements. Investigating using a mixture of motifs to model the generic binding site for datasets is promising future work.

Acknowledgements

Wolfgang Lehrach is supported by an EPSRC post-graduate student grant and Dirk Husmeier is supported by the Scottish Executive Environmental and Rural Affairs Department (SEERAD). Computing time was generously provided by Tony Travis on the SARI Beowulf cluster.

Bibliography

- D. Husmeier and K. Althoefer. Modelling conditional probabilities with network committees: how overfitting can be useful. *Neural Network World*, 8:417–439, 1998.
- C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and Wootton J. C. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
- J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Am. Stat. Assoc.*, 90(432): 1156–1170, 1995.
- T. Pawson and J. D. Scott. Signaling Through Scaffold, Anchoring, and Adaptor Proteins. *Science*, 278(5346):2075–2080, 1997.
- D. J. Reiss and B. Schwikowski. Predicting protein-peptide interactions via a network-based motif sampler. *Bioinformatics*, 20(suppl1):i274–282, 2004.
- E. Segal and R. Sharan. A discriminative model for identifying spatial cis-regulatory modules. In *RECOMB 2004 Conference Proceedings*, pages 822–834, 2004.
- E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: A probabilistic framework. In *RECOMB 2002 Conference Proceedings*, pages 263–282, 2002.
- P. Sollich and P. Krogh. Learning with ensembles: How overfitting can be useful. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 190–196. The MIT Press, 1996.
- M. Sudol and T. Hunter. New wrinkles for an old domain. *Cell*, 103:1001–1004, 2000.
- A. H. Tong, B Drees, G Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. W. V. Hogue, S. Fields, C. Boone, and G. Cesareni. A Combined Experimental and Computational Strategy to Define Protein Interaction Networks for Peptide Recognition Modules. *Science*, 295(5553):321–324, 2002.
- R. M. Twyman. *Principles of Proteomics*. BIOS Scientific Publishers, New York, 2004.
- P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Comput.*, 7(1):117–143, 1995. ISSN 0899-7667.

Improved Pattern-Driven Algorithms for Motif Finding in DNA Sequences

Sing-Hoi Sze^{1,2} and Xiaoyan Zhao¹

¹ Department of Computer Science,

² Department of Biochemistry & Biophysics,

Texas A&M University, College Station, TX 77843, USA

Abstract. In order to guarantee that the optimal motif is found, traditional pattern-driven approaches perform an exhaustive search over all candidate motifs of length l . We develop an improved pattern-driven algorithm that takes $O(4^l lk)$ time, where k is the number of sequences in the sample and l is the motif length, which is independent of the length of each sequence n for large enough l and saving a factor of n in time complexity over the original pattern-driven approach. We further extend this strategy to allow arbitrary don't care positions within a motif without much decrease in solvable values of l . Testing this algorithm on a large set of yeast samples constructed from co-expressed gene clusters reveals that most biological motifs have many invariant or almost invariant positions and these positions can be used to define the motif while ignoring the other positions. This motivates the following two-stage strategy that extends the solvable values of l substantially for the pattern-driven approach: first use an $O(2^l lkn)$ algorithm to exhaustively search over all candidate motifs allowing arbitrary don't care positions but disallowing mismatches, then refine these motifs by allowing a limited amount of flexibility to model the almost invariant positions. We demonstrate that this seemingly restrictive motif definition is sufficiently powerful by showing that the performance of this algorithm is comparable to the best existing motif finding algorithms on a large benchmark set of samples. A software program implementing these approaches (MotifEnumerator) is available at <http://faculty.cs.tamu.edu/shsze/motifenumerator>.

1 Introduction

There are roughly two types of general purpose motif finding algorithms: sample-driven approaches identify the locations of the motif occurrences directly, while pattern-driven approaches take advantage of the assumption that a motif can be specified by a central pattern and use it to reduce the search space. Although a sample-driven approach has more freedom to choose suitable statistical models (Stormo and Hartzell 1989; Lawrence et al. 1993; Bailey and Elkan 1994; Hughes et al. 2000; Workman and Stormo 2000; Thijs et al. 2001), the search space is usually so large that it is not possible to guarantee that the optimal motif is found unless the motif is very short (Leung and Chin 2005). In contrast, by assuming that a central string (in the DNA four-letter alphabet) can be used

to describe the motif, it is possible for a pattern-driven approach to perform an exhaustive search over all 4^l candidate motifs for a moderately large motif length l and guarantee that the optimal motif is found (Queen et al. 1982; Waterman et al. 1984; Staden 1989; Pesole et al. 1992; Wolfertstetter et al. 1996; van Helden et al. 1998; Tompa 1999).

A straightforward algorithm for the pattern-driven approach takes $O(4^l kn)$ time, where k is the number of sequences, n is the length of each sequence and l is the motif length, thus this strategy is feasible only for small l . By considering only candidate motifs that are at most d substitutions away from a string appearing in the sample, an extended pattern-driven approach has been proposed to reduce the number of candidate motifs from 4^l to less than $\binom{l}{d} 4^d kn$ (Waterman et al. 1984; Galas et al. 1985), and the reduction is significant when d is small relative to l . To further reduce the running time, another class of tree-based pruning techniques have been proposed (Marsan and Sagot 2000; Pavese et al. 2001; Eskin and Pevzner 2002). Fraenkel et al. (1995) proposed to combine short candidate patterns to form longer patterns, while many approaches make use of the given maximum distance d to develop heuristics that guarantee a high probability of finding the best motif (Buhler and Tompa 2002; Keich and Pevzner 2002; Price et al. 2003).

A common weakness of these approaches is that they either do not improve the worst case time complexity of the straightforward algorithm or they cannot guarantee that the optimal motif is found. In this paper, we develop an improved pattern-driven algorithm that guarantees that all statistically significant motifs are found in $O(4^l lk)$ time. This algorithm is similar to the original pattern-driven algorithm in exploring all 4^l candidate motifs of length l , but with the important difference that its time complexity is independent of the length of each sequence n (for large enough l), thus saving a factor of n in time complexity over the original pattern-driven approach. This is a significant improvement since n can be as large as 2000 and is at least 200 or 300 in many promoter finding applications. The proposed algorithm extends the power of the pattern-driven approach to find all significant motifs of length around 12 or 13 (from the original limit of around 10). It can also be adapted to handle the case when a maximum distance d is given between a motif and its occurrences.

We further extend this strategy to allow arbitrary don't care positions within a motif without much decrease in solvable values of l . This is in contrast with many previous approaches that place various constraints on the don't care positions: Rigoutsos and Floratos (1998) imposed a constraint on the density of don't care positions and developed an algorithm to identify protein motifs, while Apostolico and Parida (2004) imposed maximality and irredundancy constraints on motifs and gave an algorithm to solve the problem in cubic time when mismatches are not allowed. Although these algorithms can find very long motifs, a common weakness is that a large number of statistically significant motifs may be missed due to the constraints. Apart from these algorithms, many other approaches identify sets of composite motifs that are separated by a variable number of don't care positions, but do not allow don't care positions within each individual motif

(Marsan and Sagot 2000; van Helden et al. 2000; GuhaThakurta and Stormo 2001; Liu et al. 2001; Eskin and Pevzner 2002).

We allow arbitrary don't care positions within a motif and test our algorithm on a large set of yeast samples constructed from co-expressed gene clusters from Tavazoie et al. (1999). From the results, we observe that most biological motifs have many invariant or almost invariant positions and these positions can be used to define the motif while ignoring the other positions. This motivates the following two-stage strategy: first use an $O(2^{lkn})$ algorithm to exhaustively search over all candidate motifs allowing don't care positions but disallowing mismatches, then refine these motifs by allowing a limited amount of flexibility to model the almost invariant positions. With the much smaller exponential factor in the time complexity, this algorithm extends the solvable values of l substantially to around 20 while retaining most of the original sensitivity. We demonstrate that this seemingly restrictive motif definition is sufficiently powerful by showing that the performance of this algorithm is comparable to the best existing motif finding algorithms on a large benchmark set of samples from Tompa et al. (2005).

2 Problem Formulation

Our formulation makes a few simplifying assumptions: the central string is in the DNA four-letter alphabet and mutations occur at random positions within a motif. There are other approaches that do not have these restrictions, including those that use more general alphabets or profiles to represent a central pattern (Sinha and Tompa 2000; Price et al. 2003; Eskin 2004; Kel et al. 2004; Leung and Chin 2005) and those that take into account correlated positions within a motif (Barash et al. 2003; Zhou and Liu 2004).

We first give a formulation that allows mismatches but does not allow don't cares. Let $S = \{s_1, \dots, s_k\}$ be a sample of k sequences each of length n and let l be the length of a motif s . We put A and T together in a group and G and C together in another group. Let a be the number of A or T in s (thus $l - a$ is the number of G or C in s). Let p_1 be the probability of finding an A in the sample (which is the same as the probability of finding a T), and let p_2 be the probability of finding a C in the sample (which is the same as the probability of finding a G). The probability of s occurring with up to d substitutions at a given position of a random sequence is given by

$$p(l, a, d) = \sum_{i=0}^d \sum_{j=\max(0, a+i-l)}^{\min(a, i)} \binom{a}{j} \binom{l-a}{i-j} (1-p_1)^j p_1^{a-j} (1-p_2)^{i-j} p_2^{l-a-i+j},$$

where j counts the number of substitutions within A or T positions while i counts the total number of substitutions. To compute the p -value for s , denote the distance between s and sequence s_i by $d(s, s_i) = \min\{d(s, s') \mid s' \in s_i\}$, where s' is a string of length l appearing in s_i and $d(x, y)$ is the distance (number of substitutions) between two strings x and y of length l . Fix a maximum distance d and let k' be the number of sequences s_i with $d(s, s_i) \leq d$. The p -value of s with respect to d is given by

$$p(l, a, d, k') = \sum_{i=k'}^k \binom{k}{i} (1 - (1 - p(l, a, d))^{n-l+1})^i ((1 - p(l, a, d))^{n-l+1})^{k-i} ,$$

which is an estimate of the probability of s occurring at least once with up to d substitutions in at least k' sequences when complex correlations between overlapping patterns are ignored. Note that, for simplicity, this equation only takes into account at most one motif occurrence in each sequence. We then estimate the e -value of s with respect to d by

$$e(l, a, d, k') = 4^l p(l, a, d, k') .$$

This equation ignores differences in the nucleotide composition of motifs which may not have comparable values of a , d and k' (but still takes into account the background nucleotide distribution) and assumes that $p(l, a, d, k')$ is the probability one wishes to attain for all motifs of length l . The above equations are generalizations of the equations in Buhler and Tompa (2002) to allow for biased background distribution and some of the sequences not having a motif occurrence. We define the e -value of s to be the minimum e -value over all d . The goal of the motif finding problem is to find all motifs s with e -value below a cutoff, and the occurrences of s are defined by finding the value of d that minimizes the e -value of s and recovering all occurrences in the sample that are within distance d of s (there can be more than one occurrence in some sequences). In difference from many other approaches that assume that d is given in advance (Marsan and Sagot 2000; Pevzner and Sze 2000; Pavesi et al. 2001; Buhler and Tompa 2002; Eskin and Pevzner 2002; Keich and Pevzner 2002; Price et al. 2003), our formulation does not assume that a fixed d is given and will automatically find the best value of d for each motif s independently.

To allow for don't care positions within a motif s , let l be the length of s and l' be the number of positions within s that contain a nucleotide character (i.e., there are $l - l'$ don't care positions). A string s' of length l that appears in the sample is defined to be an occurrence of s if the total number of substitutions within these l' positions is at most d while ignoring the other $l - l'$ don't care positions. To estimate the statistical significance of a motif s , the p -value of s with respect to d is given by

$$p(l, l', a, d, k') = \sum_{i=k'}^k \binom{k}{i} (1 - (1 - p(l', a, d))^{n-l+1})^i ((1 - p(l', a, d))^{n-l+1})^{k-i} ,$$

where $p(l', a, d)$ is the same as before with l' substituting l . Since there is no need to allow don't cares at the two ends of s , the e -value of s with respect to d is given by

$$e(l, l', a, d, k') = \binom{l-2}{l'-2} 4^{l'} p(l, l', a, d, k') .$$

To allow don't cares while not allowing mismatches, simply set $d = 0$ in the above equations. Note that the notion of don't cares we use here is very different from the one in Buhler and Tompa (2002) since they used don't care positions to randomize their search procedure rather than defining motifs.

3 Algorithm When Mismatches Are Allowed

We first develop an improved pattern-driven algorithm that allows mismatches but does not allow don't cares. The original pattern-driven approach considers each candidate motif in turn and looks for its occurrences by comparing it to every string of length l in the sample. To avoid these extensive comparisons, we encode each nucleotide by two bits and create an array D of size 4^l and a queue Q of size 4^l . Our algorithm consists of two stages: the first stage computes all $d(s, s_i)$ between each candidate motif s and each sequence s_i (to be stored in D and reused for each i). We accumulate this information in another $4^l \times l$ array N which stores for each candidate motif s , the number of times that $d(s, s_i) = d$ for each d . The second stage computes the e -value of each candidate motif s from N .

The first stage iterates over each sequence s_i and starts by initializing all values in D to l (Fig. [11](#)). For each string s appearing in s_i , set $D(s)$ to 0 and insert s into Q . Repeat the following procedure that employs a depth-first search strategy: remove the first element s from Q and generate all neighbors s' of s that are one substitution away from s . For each s' , if $D(s') > D(s) + 1$, update $D(s')$ to $D(s) + 1$ and add s' to Q (Fig. [11](#)). It is easy to see that when Q becomes empty, we have $D(s) = d(s, s_i)$ for all s . Since each s appears at most once in Q and there are $O(l)$ strings that are one substitution away from s' , it takes $O(4^l l)$ time to process each sequence s_i assuming that $n < 4^l$. As the processing of each sequence s_i is completed, the values in $D(s)$ are transferred to N . The second stage uses the values in N to compute the e -value of each candidate motif s (Fig. [11](#)). Since the binomial coefficients and the probability values can be preprocessed and stored in such a way that each e -value $e(l, a, d, k')$ can be obtained in constant time and the preprocessing time is negligible (polynomial in k and l), the entire procedure takes $O(4^l l k)$ time and $O(4^l l)$ space when l is large enough. Note that the assumption $n < 4^l$ is easily satisfied: with n as large as 2000, only $l > 5$ is needed.

When implemented carefully, it is possible to store all the arrays in 4 G memory when $l = 12$ or 13 (which works on 32-bit systems). To further save memory, observe that since the values of $D(s)$ in Q are increasing, we can eliminate Q and replace it by a loop that generates neighbors s' only for those s with $D(s) = j - 1$ in iteration j . This strategy does not change the time complexity since neighbors are generated for at most 4^l strings over l iterations. Also, our approach can scan through all candidate motifs of length at most l with not much increase in running time (at most $4/3$ times longer) when compared to checking only one l . In difference from many other approaches, there is no implicit restriction on the minimum number of motif occurrences or on the maximum distance d between a motif and its occurrences. The algorithm explores all the possibilities to guarantee that the motif with the best e -value is found. When d is given, the above procedure can be used to implement the extended pattern-driven approach by stopping the first stage when the first element s in Q has $D(s) = d$ for each sequence s_i , resulting in a saving of a factor of n over the straightforward approach. Note that our neighbor generation process is similar to the one in Blanchette et al. (2002) except that their computation is based on a phylogenetic tree. Our procedure also has some

```

Algorithm MotifEnumerator( $l$ ) {
   $Q \leftarrow$  empty; for each  $(s, d)$  do {  $N(s, d) \leftarrow 0$ ; }
  for each sequence  $s_i$  do {
    for each  $s$  do {  $D(s) \leftarrow l$ ; }
    for each  $s \in s_i$  do {  $D(s) \leftarrow 0$ ; insert  $s$  to the end of  $Q$ ; }
    while  $Q$  is not empty do { remove  $s$  from the front of  $Q$ ;
      for each  $s'$  with  $d(s, s') = 1$  do {
        if  $D(s') > D(s) + 1$  then {
           $D(s') \leftarrow D(s) + 1$ ; insert  $s'$  to the end of  $Q$ ; } } }
    for each  $s$  do {  $N(s, D(s)) \leftarrow N(s, D(s)) + 1$ ; } }
  for each  $s$  do {  $k' \leftarrow 0$ ;
    for  $d \leftarrow 0$  to  $l$  do {  $k' \leftarrow k' + N(s, d)$ ;
      compute  $e(l, a, d, k')$ , where  $a$  is the number of A or T in  $s$ ; } } }

```

Fig. 1. Algorithm MotifEnumerator for finding the e -values of all candidate motifs s of length l when mismatches are allowed but don't cares are not allowed

similarity to the one in Price et al. (2003) except that our approach is exact and their approach is a heuristic.

We extend our algorithm to allow arbitrary don't care positions within a motif s . Since there is no need to allow don't cares at the two ends of s , a straightforward algorithm to enumerate all possible s of length l uses an array D of size $4^{25^{l-2}}$ to represent each s . For each sequence s_i , consider each string s' that appears in s_i and set $D(s) = 0$ for each of the 2^{l-2} possible strings s that can be generated from s' while allowing don't care positions. Then proceed in the same way as before while ignoring don't care positions during the neighbor generation process, resulting in an algorithm that takes $O(5^{l}lk)$ time and $O(5^{l}l)$ space. Alternatively, the following algorithm only needs $O(4^{l}l)$ space while having the same time complexity: for each value of l' and each way of choosing l' positions from l positions (while always choosing the two end positions), treat each string of length l with l' chosen positions as a string containing only the l' chosen positions and apply the original procedure on strings of length l' . Its time complexity can be estimated more precisely as $O(\sum_{l'=1}^l \binom{l-2}{l'-2} 4^{l'} l' k)$. When l is small (e.g., $l \leq 12$), the running time to consider motifs of length at most l with don't cares is similar to the original algorithm that considers motifs of length at most $l+1$ without don't cares, thus the modified strategy does not have a large effect on solvable values of l ($l \leq 11$ or 12 are solvable in reasonable time).

4 Test Samples

To show that our model is reasonable and the e -values are comparable over different motif lengths, we first test our algorithm MotifEnumerator on artificial samples with 20 sequences each of length 600 containing an (l, d) -motif (Pevzner and Sze 2000), which is a motif of length l with d substitutions between the motif and its occurrences. In each case, we check all candidate motifs of length at most 12 with no implicit assumption on the minimum number of motif occurrences in

a sequence or the value of d . We found that MotifEnumerator was able to find very difficult (8, 1)-, (10, 2)- and (12, 3)-motifs. In each case, the motif found was always of the correct length and the correct motif always had the best e -value.

To ensure that MotifEnumerator can identify biological motifs, we test it on a large set of yeast samples constructed from co-expressed gene clusters from Tavazoie et al. (1999) and compare our results with those in Tavazoie et al. (1999) and Hughes et al. (2000). To allow for samples having sequences of similar but unequal lengths, we use the average sequence length to approximate n . To allow for motifs to appear in the reverse complementary direction, we assume that each sequence s_i is twice as long including both the forward and the reverse complementary sequences and replace the term $n - l + 1$ by $2(n - l + 1)$ in the p -value formulas. We further preprocess each input sample by removing low complexity repeats using very simple rules. To find a set M of suboptimal motifs that are sufficiently different from each other, we first discard all motifs with e -value above a cutoff. With M initially empty, consider each remaining motif s in increasing order of e -value and repeat the following: add s to M if there are no overlaps between its occurrences and any motif occurrences already in M . This procedure finds a set of suboptimal motifs in one single run and it takes negligible time when compared to the previous stage since not many candidate motifs remain after the e -value cutoff is applied.

For each cluster in Tavazoie et al. (1999), we extract upstream sequences of length 600 resulting in a total of 30 samples, each having from 50 to 200 sequences with a nucleotide bias of around 60% A or T and 40% G or C. We run our algorithm MotifEnumerator over all motif lengths $l \leq 12$ and allow motifs to appear in the reverse complementary direction. The running time ranges from hours for the smaller samples to days for the larger samples. Table II(a) shows all strong motifs found, while Table II(b) shows a small subset of weaker motifs that are known biological motifs. Our algorithm found almost all the motifs in Tavazoie et al. (1999) and was able to identify an extra Rpn4 motif that is absent in their paper (although its e -value is not very low, it appears in more than 20 sequences). This motif was identified in Hughes et al. (2000) when a different strategy of grouping genes by common names was used to construct samples. Some of the motifs were found in a different cluster from the one specified in Tavazoie et al. (1999), including M14a (found in cluster 2) and M4 (found in cluster 16). Although they did not find any motifs in cluster 16, we found variants of M3a/M4 and M3b in cluster 16. Two motifs listed in their paper were missing from our results, including M14b and STRE that have repeating letters and were probably eliminated during the removal of low complexity repeats.

One important observation from Table II is that for almost all the motifs found, the maximum distance d that minimizes the e -value was 0. The only strong motif found in Table II(a) with $d = 1$ was Rap1, but another variant of it was also found with $d = 0$. Two motifs M1a and Rpn4 were found in Table II(b) with $d = 1$, but they are very weak and may not be distinguishable from noise. This suggests that most biological motifs can be represented accurately by invariant or almost invariant positions within the motif, which motivates an alternative

Table 1. (a) All strong motifs found by MotifEnumerator on 30 samples constructed from co-expressed yeast gene clusters from Tavazoie et al. (1999). These motifs appear in at least 10 sequences with e -value below 10^{-5} , where $cl\#$ denotes the cluster number, d denotes the maximum distance (between a motif and its occurrences) that minimizes the e -value, and don't care positions are denoted by '-'. All these motifs correspond to known biological motifs, as shown in notes. (b) A small subset of weaker motifs that are known biologically. Some of these motifs have higher e -values than over 10 other non-overlapping candidate motifs within the same run (these suboptimal motifs do not overlap with each other). M3a/M4 and Cbf1p appear in less than 10 sequences.

(a)				(b)					
cl#	motif	d	e -value	notes	cl#	motif	d	e -value	notes
1	acatccgtacat	1	1.12e-25	Rap1	1	tttctcact-t	1	3.18e-02	M1a
1	accca-acat-t	0	6.31e-10	Rap1	2	ttcttg	0	8.47e-05	SCB
2	acgcgt-a	0	2.24e-22	MCB	2	tggcaaatg	1	2.35e-03	Rpn4
2	tttcgcg	0	2.16e-09	SCB/M14a	3	tgaaaaatttt	0	1.41e-05	M3a/M4
3	gatgagatgag	0	1.44e-16	M3b	7	c-aaa--gg-aa	0	7.99e-04	ECB
3	cgatgagc	0	9.76e-08	M3b	30	gtcacgtgc	0	1.20e-03	Cbf1p
7	tttcc-aa--g	0	5.56e-08	ECB					
8	ttcttg	0	7.20e-06	SCB					
16	gcatgag-t	0	1.15e-16	M3b					
16	tgaaaaattt	0	7.58e-06	M3a/M4					
30	gccacag	0	2.99e-06	Met31/32p					

formulation that disallows mismatches when arbitrary don't cares are allowed. With this restriction, the problem becomes easier to solve and longer motifs can be considered. To improve the sensitivity in finding plausible motif occurrences, a limited number of mismatches can be allowed by adding a post-processing step to refine the initial motifs.

5 Algorithm When Mismatches Are Not Allowed

We first give an algorithm that takes $O(lkn)$ time and space when both mismatches and don't cares are not allowed. Under these assumptions, each string s of length l that appears in the sample represents a candidate motif. We store these strings in a tree T of height l so that each s is represented by a path of length l from the root. Each internal node t of T can have at most four children $t.c$, one for each character c of the DNA alphabet, with the path from the root to t representing a prefix of one or more motifs; while each leaf node t of T represents a unique motif s , with $t.k'$ denoting the number of sequences that s occurs in (only at most one occurrence is counted in each sequence) and $t.i$ denoting the sequence number of the previous occurrence of s during the tree construction (Fig. 2). To allow for arbitrary don't care positions, for each value of l' and each way of choosing l' positions from l positions (while always choosing the end positions), treat each string of length l with l' chosen positions as a string containing only the l' chosen positions and build a tree T of height l' .

```

Algorithm MotifEnumerator( $l$ ) {
   $T \leftarrow$  root with no children;
  for each sequence  $s_i$  do {
    for each  $s \in s_i$  do {  $t \leftarrow$  root of  $T$ ;
      for  $j \leftarrow 1$  to  $l$  do {  $c \leftarrow$   $j$ th character of  $s$ ;
        if  $t.c$  does not exist then {  $t.c \leftarrow$  new node with no children;  $t \leftarrow t.c$ ;
          if  $j = l$  then {  $t.k' \leftarrow 0$ ;  $t.i \leftarrow -1$ ; } }
          else {  $t \leftarrow t.c$ ; } }
        if  $t.i \neq i$  then {  $t.k' \leftarrow t.k' + 1$ ;  $t.i \leftarrow i$ ; } } }
    for each leaf  $t$  of  $T$  do {
      compute  $e(l, a, 0, t.k')$ , where  $a$  is the number of A or T in the motif in  $t$ ; } }

```

Fig. 2. Algorithm MotifEnumerator for finding the e -values of all candidate motifs s of length l when both mismatches and don't cares are not allowed

The entire procedure takes $O(2^l lkn)$ time and $O(lkn)$ space, thus by disallowing mismatches, we extend the solvable values of l to around 20 (from around 12 when mismatches are allowed). Also, our approach can scan through all candidate motifs of length at most l with not much increase in running time (at most twice longer) when compared to checking only one l .

Although the above procedure can be quite successful in identifying core motif occurrences, the requirement that each occurrence must be exactly the same except for the don't care positions is very strict, thus it is likely that some motif variants are missed. We use the following strategy to allow for a limited number of mismatches while avoiding the introduction of many false positives: let s be a motif of length l with m occurrences o_1, \dots, o_m each of length l (there can be more than one occurrence in some sequences). We construct a refined motif s' as follows: for each position j , if there exists a nucleotide character c such that its total frequency at the j th position within the m occurrences is more than $m/2$, set the j th character of s' to c , otherwise set it to a don't care character (note that c is uniquely defined if it exists). Let $d' = \max\{d(s', o_i) \mid 1 \leq i \leq m\}$, where the don't care positions in s' are ignored when computing distances. We define the occurrences of s' to be all strings of length l that appear in the sample and are within distance d' of s' . Note that this new set of occurrences of s' must include the original occurrences of s .

6 Benchmark Test Samples

We test this new version of MotifEnumerator on a large benchmark set of samples from Tompa et al. (2005), each having up to 35 sequences with sequence lengths ranging from 500 to 3000. Since many biological motifs in the test set contain moderately repeating patterns, we use a less extensive procedure than before to remove low complexity repeats that include single-nucleotide repeats of length at least six, two-nucleotide repeats with at least four repeating units, and three-nucleotide repeats with at least three repeating units, with no mismatches allowed within the repeats. We run MotifEnumerator over all motif

lengths $l \leq 20$ and look for motifs only on the forward strand. In each case, the refined occurrences of the top motif with e -value below 1.0 are used for evaluation (it is possible that no motif is found). The running time ranges from hours for the smaller samples to days for the larger samples.

Table 2 shows the performance of MotifEnumerator on both the mixed set of samples that was assessed in Tompa et al. (2005) and on the original three sets of samples of type real, generic and markov from which the mixed set is derived but were not assessed in Tompa et al. (2005). On the mixed set, the overall performance of MotifEnumerator (with $nCC=0.067$) was roughly comparable to algorithms assessed in Tompa et al. (2005) that had overall performance ranging from above average to near-best, including AlignACE (Hughes et al. 2000) with $nCC=0.068$, MotifSampler (Thijs et al. 2001) with $nCC=0.068$, MEME (Bailey and Elkan 1994) with $nCC=0.073$, Oligo/Dyad (van Helden et al. 1998; van Helden et al. 2000) with $nCC=0.071$, and ANN-Spec (Workman and Stormo 2000) with $nCC=0.074$. Only two algorithms definitely performed much better, including YMF (Sinha and Tompa 2000) with $nCC=0.084$ and Weeder (Pavesi et al. 2001) with $nCC=0.156$. Within the mixed set, MotifEnumerator followed a similar trend as most other algorithms, with better performance on samples of type generic and markov and worse performance on samples of type real. In particular, on samples of type real, MotifSampler (Thijs et al. 2001) with $nCC=0.076$ and Weeder (Pavesi et al. 2001) with $nCC=0.077$ performed best among all the assessed algorithms, while YMF (Sinha and Tompa 2000) with $nCC=0.013$ performed much worse than MotifEnumerator with $nCC=0.046$. Overall, Weeder (Pavesi et al. 2001) had the best performance that was much higher than all the other assessed algorithms. When the samples were categorized by the organism from which the upstream sequences are obtained, MotifEnumerator also followed a similar trend as most other algorithms, with the best performance on yeast samples, medium performance on human and mouse samples and worst performance on fly samples.

We also analyze the performance of MotifEnumerator on the original three sets of samples of type real, generic and markov from which the mixed set is derived. Each of these original sets contains about the same number of samples as the entire mixed set (Table 2). The most noticeable advantage of MotifEnumerator is that similar overall performance was obtained across all these original sets with distinct background types and thus MotifEnumerator does not seem to be affected much by differences in the background sequences. Also, there was a significant increase in the performance of MotifEnumerator on the fly samples within the real set, which is mainly due to a strong result on the dm01r sample (this sample was not assessed in Tompa et al. (2005)). Interestingly, Tompa et al. (2005) also reported that MotifSampler (Thijs et al. 2001) had similar performance over different background types within the mixed set and SeSiMCMC (Favorov et al. 2005) had strong performance on the fly samples within the mixed set (although SeSiMCMC (Favorov et al. 2005) had weak overall performance).

Table 2. Performance of MotifEnumerator on benchmark test samples from Tompa et al. (2005) when arbitrary don't care positions are allowed but mismatches are not allowed. Each entry represents the nucleotide-level correlation coefficient (nCC) computed by comparing the refined occurrences of the top motif returned from MotifEnumerator (if one exists) to the known annotation in each sample and treating a subset of samples as if it was a single large sample. Each row represents a set of 56 samples (except for the set of type real, which contains 52 samples). Each set of type real, generic or markov contains motifs corresponding to one transcription factor with a particular type of background sequences. Tompa et al. (2005) did not perform assessments directly on these sets, but constructed another set of type mixed with 56 samples by picking one background type for each transcription factor (out of a total of two or three possibilities) so that samples within this set may have different background types. Assessments were performed only on this mixed set in Tompa et al. (2005), which corresponds to the row and the column labeled mixed, while ignoring the other 108 samples from the original sets. Each set is further subdivided into four subsets according to the organism from which the upstream sequences are obtained (except for the mixed subset, which contains samples of a particular type within the entire mixed set).

	mixed	fly	human	mouse	yeast	overall
mixed		-0.010	0.040	0.043	0.238	0.067
real	0.046	0.075	0.025	0.058	0.214	0.063
generic	0.091	-0.010	0.014	0.046	0.335	0.065
markov	0.065	-0.010	0.051	0.052	0.188	0.073

7 Discussion

Since allowing mismatches may still provide better sensitivity in some cases, both variants of MotifEnumerator are useful in different situations. The main advantage of allowing mismatches is that a one-step process can be used to guarantee that the optimal motif is found while automatically allowing appropriate variations if the resulting statistical evaluation is favorable. The time complexity of our algorithm contains an exponential factor and is independent of the length of each sequence n for large enough l . Thus it is useful in most situations when the goal is to identify the conserved core region of a promoter.

When mismatches are not allowed, the search space is much smaller and it becomes possible to develop an algorithm with a much smaller exponential factor in the time complexity that only needs polynomial space instead of exponential space, thus allowing longer motifs to be considered while still guaranteeing that the optimal motif pattern is found. Although the tests above show that the algorithm is not very fast when l is around 20, it is extremely fast when l is small. For example, it takes seconds to run the algorithm for the smaller samples in Tompa et al. (2005) and minutes to hours for the larger samples over $l \leq 10$ or 12. To avoid missing important motif occurrences, an additional step has been introduced to find plausible motif occurrences while allowing limited mismatches. Although we have used a strict definition in this step to avoid introducing many false positives, it is also possible to use less strict definitions to allow more occurrences to be identified. In spite of the seemingly restrictive

motif definition in disallowing mismatches initially, our algorithm does not seem to lose much sensitivity when compared to most other algorithms assessed in Tompa et al. (2005) that use more general motif models. Only Weeder (Pavesi et al. 2001) consistently performed much better than MotifEnumerator in almost all situations.

To further improve the algorithms, it may be desirable to allow a small amount of overlaps among suboptimal motif occurrences to avoid missing motifs. It is also important to develop more accurate statistical formulas for samples that do not have sequences of similar lengths and for motifs with more than one occurrence per sequence. This has to be done very carefully since assigning scores that correspond to many occurrences on a sequence may not necessarily lead to an increase in sensitivity due to the larger flexibility that allows many other candidate motifs to have better scores. To further improve performance, it may be desirable to incorporate genome-specific information by using the overall genome nucleotide distribution, probably only in the non-coding regions, to serve as the background distribution. In many situations, there may be a need to find motifs that are significant in one sample but not in the other. This can be addressed by extracting motifs in one sample that have a good likelihood ratio with respect to another negative sample.

Acknowledgments

This work was supported by NSF grants CCR-0311590 and DBI-0421815.

References

- Apostolico, A., Parida, L.: Incremental paradigms of motif discovery. *J. Comp. Biol.* **11** (2004) 15–25
- Bailey, T.L., Elkan, C.P.: Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. 2nd Int. Conf. Intelligent Systems Mol. Biol. (ISMB'1994)* 28–36
- Barash, Y., Elidan, G., Friedman, N., Kaplan, T.: Modeling dependencies in protein-DNA binding sites. *Proc. 7th Ann. Int. Conf. Res. Comp. Mol. Biol. (RECOMB'2003)* 28–37
- Blanchette, M., Schwikowski, B., Tompa, M.: Algorithms for phylogenetic footprinting. *J. Comp. Biol.* **9** (2002) 211–223
- Buhler, J., Tompa, M.: Finding motifs using random projections. *J. Comp. Biol.* **9** (2002) 225–242
- Eskin, E.: From profiles to patterns and back again: a branch and bound algorithm for finding near optimal motif profiles. *Proc. 8th Ann. Int. Conf. Res. Comp. Mol. Biol. (RECOMB'2004)* 115–124
- Eskin, E., Pevzner, P.A.: Finding composite regulatory patterns in DNA sequences. *Bioinformatics* **18** (2002) S354–363
- Favorov, A.V., Gelfand, M.S., Gerasimova, A.V., Ravcheev, D.A., Mironov, A.A., Makeev, V.J.: A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics* **21** (2005) 2240–2245

- Fraenkel, Y.M., Mandel, Y., Friedberg, D., Margalit, H.: Identification of common motifs in unaligned DNA sequences: application to *Escherichia coli* Lrp regulon. *Comp. Appl. Biosci.* **11** (1995) 379–387
- Galas, D.J., Eggert, M., Waterman, M.S.: Rigorous pattern-recognition methods for DNA sequences. Analysis of promoter sequences from *Escherichia coli*. *J. Mol. Biol.* **186** (1985) 117–128
- GuhaThakurta, D., Stormo, G.D.: Identifying target sites for cooperatively binding factors. *Bioinformatics* **17** (2001) 608–621
- Hughes, J.D., Estep, P.W., Tavazoie, S., Church, G.M.: Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.* **296** (2000) 1205–1214
- Keich, U., Pevzner, P.A.: Finding motifs in the twilight zone. *Bioinformatics* **18** (2002) 1374–1381
- Kel, A., Tikunov, Y., Voss, N., Wingender, E.: Recognition of multiple patterns in unaligned sets of sequences: comparison of kernel clustering method with other methods. *Bioinformatics* **20** (2004) 1512–1516
- Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262** (1993) 208–214
- Leung, H.C., Chin, F.Y.: Finding exact optimal motifs in matrix representation by partitioning. *Bioinformatics* **21** (2005) SII86–92
- Liu, X., Brutlag, D.L., Liu, J.S.: BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Sym. Biocomp. (PSB'2001)* 127–138
- Marsan, L., Sagot, M.-F.: Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J. Comp. Biol.* **7** (2000) 345–362
- Pavesi, G., Mauri, G., Pesole, G.: An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* **17** (2001) S207–214
- Pesole, G., Prunella, N., Liuni, S., Attimonelli, M., Saccone, C.: WORDUP: an efficient algorithm for discovering statistically significant patterns in DNA sequences. *Nucleic Acids Res.* **20** (1992) 2871–2875
- Pevzner, P.A., Sze, S.-H.: Combinatorial approaches to finding subtle signals in DNA sequences. *Proc. 8th Int. Conf. Intelligent Systems Mol. Biol. (ISMB'2000)* 269–278
- Price, A., Ramabhadran, S., Pevzner, P.A.: Finding subtle motifs by branching from sample strings. *Bioinformatics* **19** (2003) SII149–155
- Queen, C., Wegman, M.N., Korn, L.J.: Improvements to a program for DNA analysis: a procedure to find homologies among many sequences. *Nucleic Acids Res.* **10** (1982) 449–456
- Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics* **14** (1998) 55–67
- Sinha, S., Tompa, M.: A statistical method for finding transcription factor binding sites. *Proc. 8th Int. Conf. Intelligent Systems Mol. Biol. (ISMB'2000)* 344–354
- Staden, R.: Methods for discovering novel motifs in nucleic acid sequences. *Comp. Appl. Biosci.* **5** (1989) 293–298
- Stormo, G.D., Hartzell, G.W.: Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci. USA* **86** (1989) 1183–1187
- Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. *Nature Genet.* **22** (1999) 281–285

- Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouzé, P., Moreau, Y.: A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17** (2001) 1113–1122
- Tompa, M.: An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *Proc. 7th Int. Conf. Intelligent Systems Mol. Biol. (ISMB'1999)* 262–271
- Tompa, M., Li, N., Bailey, T.L., Church, G.M., De Moor, B., Eskin, E., Favorov, A.V., Frith, M.C., Fu, Y., Kent, W.J., Makeev, V.J., Mironov, A.A., Noble, W.S., Pavese, G., Pesole, G., Régnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., Zhu, Z.: Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotech.* **23** (2005) 137–144
- van Helden, J., André, B., Collado-Vides, J.: Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* **281** (1998) 827–842
- van Helden, J., Rios, A.F., Collado-Vides, J.: Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.* **28** (2000) 1808–1818
- Waterman, M.S., Arratia, R., Galas, D.J.: Pattern recognition in several sequences: consensus and alignment. *Bull. Math. Biol.* **46** (1984) 515–527
- Wolfertstetter, F., Frech, K., Herrmann, G., Werner, T.: Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm. *Comp. Appl. Biosci.* **12** (1996) 71–80
- Workman, C.T., Stormo, G.D.: ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. *Pac. Sym. Biocomp. (PSB'2000)* 467–478
- Zhou, Q., Liu, J.S.: Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics* **20** (2004) 909–916

Annotation of Promoter Regions in Microbial Genomes Based on DNA Structural and Sequence Properties

Huiquan Wang and Craig J. Benham

UC Davis Genome Center, University of California
One Shields Avenue, Davis, CA 95616
{cjbendam, hqwang}@ucdavis.edu

Abstract. Understanding the attributes that confer promoter activity is essential for gene regulation, gene prediction and sequence annotation. It is a challenging problem to detect promoter regions, either *in silico* or by experiments. In this report, we show that the stress induced DNA duplex destabilization sites (SIDD) in prokaryotic genomes under negative superhelical stresses, as occurs *in vivo*, are closely associated with specific promoter regions. When compared with DNA curvature, deformability, thermostability or sequence motif scores within the -10 region, SIDD is the most informative DNA property of promoter regions in the *E. coli* K12 genome. Our method using SIDD as a sole predictor performs better than other promoter prediction programs in detecting promoter sequences in *E. coli* or *Bacillus subtilis*. We show that, by combining SIDD properties with -10 motif scores in a linear discrimination function, one can achieve better than 80% accuracy in predicting promoter sequences.

1 Introduction

As the number of completely sequenced microbial genomes grows, the need for efficient annotation tools becomes more acute. Gene-finding programs such as GeneMark or Glimmer [1, 2] can predict protein coding regions at a generally high level of accuracy. However, there also are genes encoding rRNA, tRNA and small non-coding RNAs in prokaryotic genomes, which these methods may not always find. The precise locations of translation and transcription start sites also need to be identified. Better understanding of the attributes associated with promoters, in addition to shedding light on the basic mechanisms by which they function, will also assist in identifying these sites within genomic sequences.

Promoter prediction in prokaryotic genomes presents unique challenges owing to their organizational properties. First, gene densities are very high in prokaryotes – 89% of base pairs in the *E. coli* genome are in open reading frames (ORFs). Neighboring genes may have very short intergenic regions; in some cases their coding regions even overlap. Further, the operon structure, in which multiple genes are transcribed as a single transcription unit, means that not all genes require their own promoters. In order to thrive in different environments, genomic sequences are highly adaptive within a genome and highly diversified across genomes. This makes it difficult to detect conserved regulatory sites within and across genomes by sequence homology. These factors have complicated the search for the determinants of promoter activity in prokaryotes.

Prokaryotic promoters are known to contain conserved sequence motifs, which may be represented either as consensus sequences or by position-specific score matrices (PSSMs) [3]. For example, most *E. coli* K12 promoters contain approximately conserved sequence elements in their -35 and -10 regions [4-6]. The -10 motif is essential for transcription initiation, while -35 motif is dispensable for some promoters. Other sequence features of *E. coli* K12 promoters include the A+T rich so-call “UP element” located at about -50bp [7]. Most of the promoter prediction programs thus far developed search sequences for conserved -10 motifs, and in some cases also include -35 motifs [8, 9]. These methods commonly suffer from high false positive rates.

The melting of double strand DNA in the promoter regions is a critical step in transcription initiation for both prokaryotes and eukaryotes. *In vivo*, DNA is generally negatively supercoiled. The untwisting torsional stress this imposes can destabilize the DNA duplex in specific regions of a genome, and thereby facilitate local strand opening. We have developed methods to analyze this stress-induced DNA duplex destabilization. For a specified level of superhelicity, this calculates two quantities for each base pair - the probability of its opening, and the incremental free energy $G(x)$ needed to force it to be always open under these conditions [10, 11]. A small number of specific sites in genomic DNA are predicted by SIDD analysis to have a high propensity to melt under normal physiological conditions. We have demonstrated that these SIDD sites in the *E. coli* K12 genome are statistically significantly associated with intergenic regions that are known or inferred to contain promoters. It is found that many - but not all - documented promoters contain a strong SIDD site. Further, SIDD sites also occur at frequencies much below expectation in coding regions [12]. This pattern of SIDD site distribution has recently been confirmed to occur in many other prokaryotic genomes [13]. This suggests that SIDD properties may be used to investigate promoter-containing regions in prokaryotic genomes.

In this report we show that SIDD is a distinct structural property of promoter regions that cannot be captured by sequence analysis. When compared with other known DNA structural properties and -10 motif scores, SIDD properties are found to be the best discriminator for distinguishing promoters from non-promoters. When SIDD was either the sole predictor or was combined with other features in a promoter prediction program, significant improvements of sensitivity and specificity were achieved. We anticipate that this approach may be developed into a method to predict promoter locations in sequenced prokaryotic genomes.

2 Results

SIDD is a distinct structural property in promoter regions that cannot be captured by sequence conservation. We randomly selected 500 documented transcription start sites (TSS) from the total of 927 annotated in the Regulon database for the *E. coli* K12 genome. The regions immediate upstream of these TSSs were considered to be promoters. We calculated the average destabilization free energy $G(x)$ for each location within the 1001 bp sequence centered on these TSSs. For comparison we performed the same analysis on three other sets, each containing 500 sequences that are 1001 bp long. The first set starts at the TSSs, and hence contains

transcribed regions. The second set consists of 500 randomly selected sequences, each 1001 bp long, and centered on an intergenic region separating convergently oriented genes. This latter set, which we call CON, may contain ORFs but is inferred not to contain promoters. In the third set, the *E. coli* K12 genome was randomly shuffled first and subjected to SIDD calculation. A set of 500 random sequences together with their SIDD profiles were then chosen for this analysis.

The results of this analysis are shown in Figure 1A. On average, the sequences containing promoters are more destabilized than either their coding sequences or the CON sequences that may not contain promoters. These results agree with the results of our previous analysis, which showed strong SIDD sites to be statistically significantly associated with divergent or tandem intergenic regions that contain promoters or other transcriptional regulatory elements. The lowest average value of $G(x)$ occurs around position -49, a region previously identified as the “UP element” in some *E. coli* K12 promoters. (The TSS is placed at position 0, so all other positions are reported relative to this site.) Statistical analysis shows the regions between positions -174 and +57 are significantly destabilized ($p < 0.001$) when compared with non-promoter sequences. These regions are also significantly destabilized when compared with the random sequences. In what follows we designate the 100 bp sequences from positions -80 to +19 as promoter sequences.

The 500 promoter sequences were aligned at their TSSs, and their sequence conservation was measured by Shannon’s entropy. This is shown in Figure 1B. As expected, a slight entropy decrease (which corresponds to increased sequence conservation) was observed between positions -7 and -12, corresponding to their -10 regions. No similar conservation was noted in the -35 regions. It is known that the -35 motif is dispensable for so-called “extended -10” sigma 70 promoters. Strikingly, the region around -49 bp where the maximal destabilization was found also shows no increase in sequence conservation. This indicates that SIDD properties and -10 motifs are fundamentally different attributes of promoter sequences; one is tied directly to sequence but the other is not.

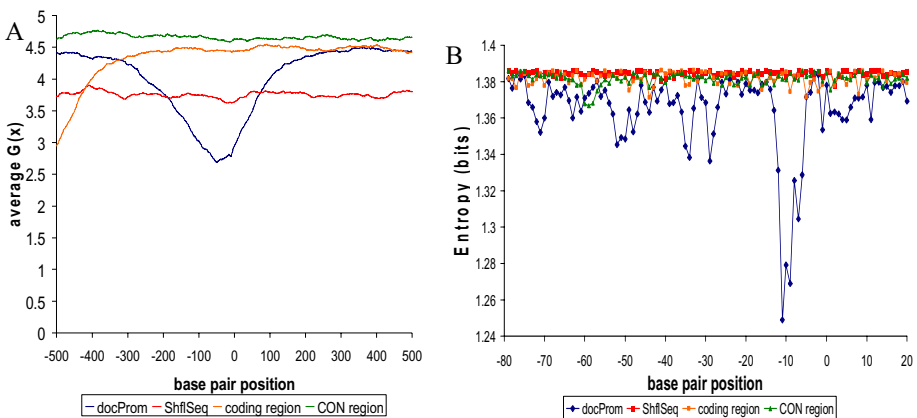


Fig. 1. A. Promoter regions in *E. coli* K12 are generally more destabilized than coding or CON regions. B. -10 regions in promoter sequences are conserved while other regions are not in *E. coli* K12.

A recent paper that presented a promoter prediction method based on thermostability in promoter regions reported that the region from -20 to -6 was much less stable than were other, non-promoter locations [14]. It is possible that the decreased thermostability in this region were partly contributed by the -10 conserved sequences, which is A+T-rich.

This result emphasizes the fact that SIDD properties are fundamentally different from thermostability. SIDD does not depend only on the local thermal properties of the local DNA sequence. The energies that govern SIDD are the differences between the energy cost for base pair melting under the specified superhelical stress, and the energy benefit of the relaxation this melting provides. The thermal energy only relates to the cost half of this relationship. The superhelical stresses couple together all the base pairs that experience them, so whether SIDD melting occurs at any specific site depends on how that site competes with all other sites that feel this stress. This means a site can open at one level of stress, then reclose coupled to opening elsewhere as the stresses are increased. (See Fig 2 of the reference [11], where both sites have the same thermodynamic stability.) This type of complicated, nonlinear behavior does not occur in thermal melting, and cannot be predicted only from the thermal properties of the sequence.

SIDD energy levels are bimodally distributed among promoter sequences. We measured two SIDD attributes for each of the training sets of promoter and non-promoter sequences. The first attribute is the sum Σ of the $G(x)$ values for all base pairs in the region, and the second is G_m , the minimum value attained by $G(x)$ in the region. The distributions of these quantities over the two sets are shown in Figure 2. The majority of the non-promoter sequences have high values of both Σ and G_m , indicating that they remain stable under superhelical stress. However, the distributions of both parameters at promoters are bimodal. It appears that two sub-populations of promoter sequences can be distinguished according to their SIDD properties, one group highly destabilized and the other less so.

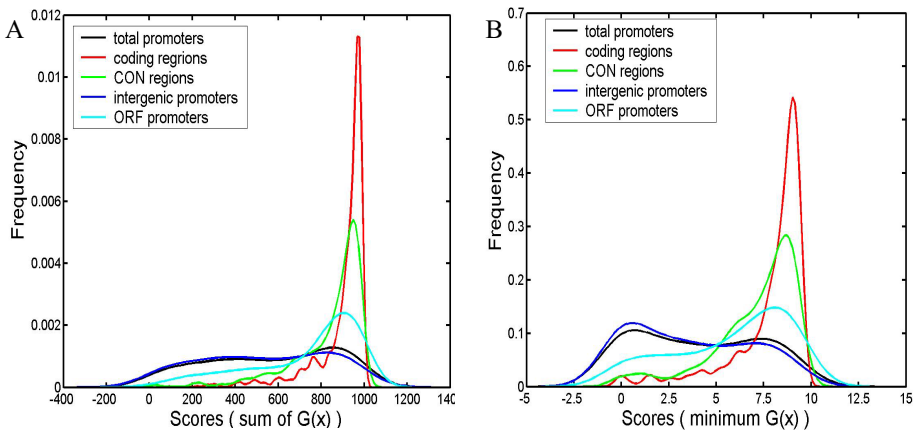


Fig. 2. The distribution of SIDD in promoter, coding, and CON regions. (A) sum of $G(x)$; (B) minimum $G(x)$.

Because we previously showed that strong SIDD sites are closely associated with promoter-containing intergenic regions and avoid coding regions [12], we wanted to test whether these two subgroups of promoters are from these two different locations. More than 80% of the 927 experimentally characterized TSSs are located in intergenic regions. We separated this set into intergenic promoters and ORF promoters, and examined their SIDD characteristics separately. As shown in Figure 2, there may be a slight enrichment of ORF promoters in the non-destabilized population, but some bimodal character is preserved in each of these promoter sets.

The observed bimodal distribution of SIDD properties in promoters may reflect the complexity of transcriptional regulation, suggesting that destabilization may be needed in initiating transcription from some promoters, but not others. One can imagine that SIDD in highly destabilized promoters may be involved in the mechanism of open complex formation. However, one cannot rule out the possibility that SIDD may also be involved in regulating more stable promoters. In this analysis we confined our attention to the 100 bp sequence from -80 bp to +19 bp. But SIDD sites further upstream are known to be involved in the IHF-mediated transcriptional activation of the promoter governing expression of the *E. coli* *ilvGMEDA* operon [15]. In the absence of IHF binding, negative superhelicity opens the SIDD region. IHF binding forces this region back to B-form, which causes the next most easily destabilized site to open, which in this case is in the -10 region. In this case SIDD plays an important role in the mechanism of gene expression even though the regulatory SIDD site is not at the promoter.

SIDD is more capable than other structural or sequence properties of distinguishing promoters from non-promoter sequences in *E. coli* K12. Other studies have suggested that several other structural parameters also behave differently between promoters and non-promoters. These include DNA intrinsic curvature, protein induced deformability, and thermodynamic stability [16-18]. To compare these with SIDD properties, we measured the sums and minimum values of each of these variables over our test sets. The distributions of these parameters in promoter sequences and non-promoter sequences are shown in Figure 3ABC. Overall, promoter regions tend to be slightly more curved, more easily deformed by protein binding, and less stable under thermal fluctuation than non-promoter regions. However, the differences are much less dramatic than those seen for SIDD.

Position-specific score matrices (PSSMs) are frequently used to represent and search sequences for conserved motifs, such as protein binding sites. PSSM methods that use either the -10 motif or motifs from both the -10 and -35 regions have been used to detect sigma factor binding sites, presumably as signals for promoters. By aligning promoter sequences at their TSSs, we derived a log-odds PSSM for the -10 motif (from -6 to -13) in the promoters according to the description in [3]. When this PSSM is used to search promoter and non-promoter sequences, we find that promoters contain higher densities of high-scoring PSSM motifs than do non-promoters (figure 3D and data not shown). As there are substantial numbers of these motifs in both sets, this suggests that the exact location of the -10 motif may not be useful for inferring either transcription start sites or promoter locations.

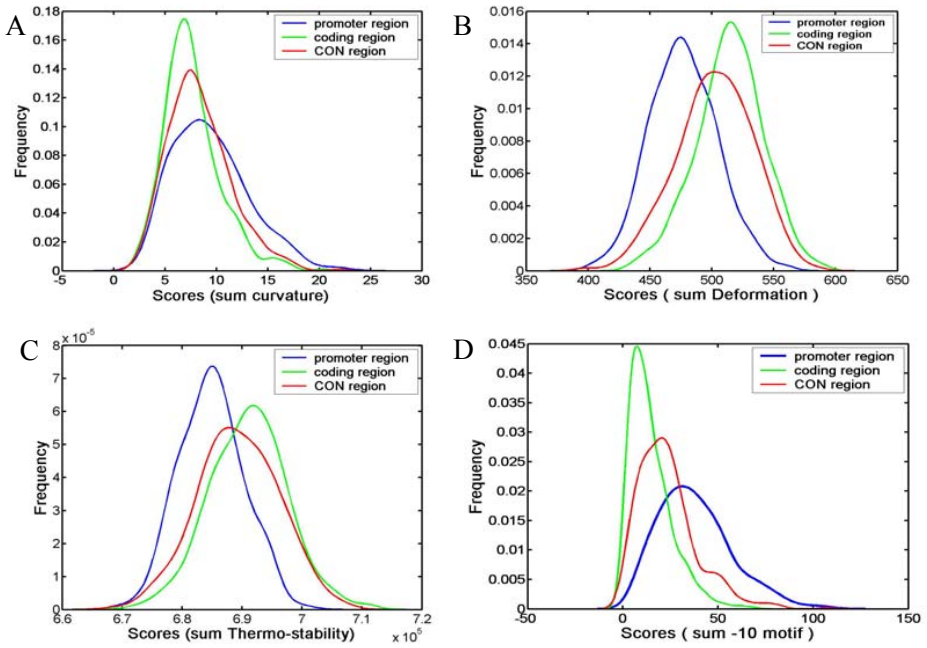


Fig. 3. Distributions of the sums of the scores for curvature (A), deformability (B), thermostability (C) and sum of -10 motif scores (D) in promoter regions, coding regions and CON regions

Table 1. SIDD is the most distinct variable that differentiates promoter from non-promoter sequences

	Promoter region vs	
	Coding region	CON region
SIDD	1.0308e-076 ^a /4.0961e-072 ^b	1.0398e-046/2.5736e-044
Curvature	2.4277e-15/8.0973e-004	5.3170e-005/0.0774
Deformation	7.116e-063/1.0	1.2783e-31/0.8567
Thermo-Stability	5.5028e-042/1.0	1.0527e-014/1.0
-10 motif	1.1882e-74 ^c	2.6299e-30

Each value in the table is the probability that the two distributions are the same, as found using the Kolmogorov-Smirnov two sample test.

a, sum of the values of the variables in the sequences; b, minimum value of the variable in the sequences c, sum of the -10 motif scores of the sequences.

The distributions of each individual variable (SIDD, curvature, deformability, thermodynamic stability, and -10 motif scores) were examined for promoter and non-promoter sequences in each of two cases – first using the summed variable and then (except for -10 scores) its extreme value in the region. The comparison non-promoter regions were chosen to be either coding regions or CON regions. The statistical significances of the differences were calculated in each case using the Kolmogorov-Smirnov test. The results of these statistical analyses are shown in Table 1. In all cases the distributions of the summed variables show statistically significant differences. Among these, the SIDD property shows the highest significance level. The distribution of extreme values remains highly significant for SIDD, but is much less so for the other parameters. In fact, only the curvature difference retains any significance, and that only when comparing promoters with coding regions. SIDD is the most informative variable for differentiating promoter from non-promoter sequences, and -10 motif scores are also informative.

SIDD alone outperforms other programs in detecting promoter sequences in *E. coli* K12 or *B. subtilis*. Since the SIDD profile can be directly calculated from primary sequence, it can be advantageous to use it as a predictor when other prior knowledge is limited. One sorts sequences into promoters-containing and non-promoter bins according to the values of the parameters Σ and/or G_m they attain. This is done for all sequences in positive and negative training sets for each pair of values of Σ and G_m . We calculate the true and false positive rates in each case. In this way a set of values of Σ and G_m were found that optimally discriminate promoter regions from non-promoter sequences.

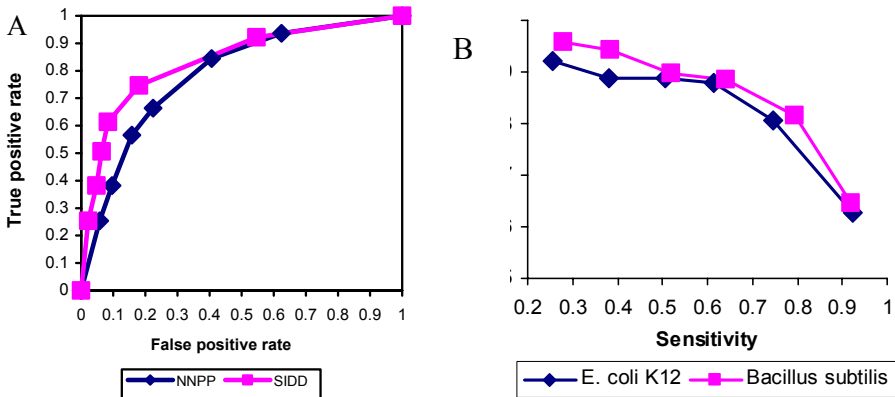


Fig. 4. (A) ROC comparison of SIDD vs NNPP performance identifying promoters in *E. coli* K12. (B) SIDD achieves comparable performance discriminating promoter and non-promoter sequences in *E. coli* K12 and *Bacillus subtilis*.

We compared the performances on these test sets of this optimized SIDD predictor and the publicly available NNPP promoter prediction program. Although NNPP was originally developed to predict core promoter regions in the *Drosophila melanogaster* genome [19], it was also trained on a set of documented promoter sequences in *E. coli*

K12. NNPP is a neural network-based computer program that uses a time-delay architecture to incorporate structural and compositional properties of promoter sequences. By setting its threshold of stringency between 0.1 and 0.9, we obtained a range of predictions regarding the test sequences. The true positive and false positive rates were calculated at each threshold. Since the SIDD method detects promoter-containing regions without pinpointing the TSS while NNPP predicts TSSs, care must be taken to calculate true and false positive rates in equivalent manners. If NNPP predicts that a sequence contains a TSS, no matter where it is, that sequence is considered to contain a promoter. The performance characteristics of the SIDD predictor and NNPP were compared using a ROC (receiver operating characteristic) curve, which shows the trade off between sensitivity and specificity. The area under the ROC is a convenient way of comparing classifiers. The better the predictor the more the curve moves towards the vertical axis. As we can see from Figure 4A, by this criterion SIDD has better predictive power than NNPP. At a given false positive rate, SIDD predicted more true positives than NNPP. For example, SIDD correctly predicted 74.6% of the real promoters with an 18% false positive rate. When NNPP correctly predicted 66.4% of the real promoters it had at 22.4% false positive rate. We note, however, that the number of *E. coli* K12 promoter sequences used to train NNPP was about half the size of the set we used in our study.

The pattern of SIDD distribution found in *E. coli* K12 also occurs in other prokaryotic genomes. So we also applied our methods to *Bacillus subtilis*, the only other prokaryote from a different phylum that has extensive experimental annotation of promoters. In this organism the most extreme level of average destabilization also occurs at position -49 relative to the TSS. The sequence at this site is not as conserved as that of -10 regions. A bimodal distribution of SIDD properties is found in *Bacillus subtilis* promoter regions, just as it is in *E. coli* K12. (Data not shown) When SIDD alone was used to differentiate promoter from non-promoter sequences in *Bacillus subtilis*, it achieve an 80% true positive rate with an 18% of false positive rate, comparable to its performance in *E. coli* K12 (Figure 4B). Despite of the different nucleotide compositions of the genomes of *E. coli* K12 and *Bacillus subtilis*, and the large evolutionary distance separating them, SIDD consistently predicts promoter and non-promoter sequences in both organisms with comparably high true positive rate and low false positive rates. Thus, SIDD characteristics may be capable detecting promoter sequences in many prokaryotic genomes.

A recent paper using thermostability as a predictor also claimed that their method was likely to be applicable across different microorganisms[14]. Indeed, both our method and theirs were tested on the same source - experimental TSSs from *E. coli* and *Bacillus subtilis*. Since their program was not publicly available, we tried to compare the performances of both methods by plotting the figure 4B from our data similar to the Figure 9 of their paper, which showed the prediction accuracy of their method. The definitions of the precision and sensitivity in figure 4B are the same as the ones defined there. As the level of sensitivity increases from 20% to 90%, the prediction precision of the method based on thermostability decreased dramatically from about 72% to about 37% for *E. coli*, and from about 82% to 27% for *B. subtilis*, respectively. On the contrary, the prediction precision of our method at all level of sensitivity remained above 62% for both *E. coli* and *B. subtilis*. As can be seen from Figure 4B here and the Figure 9 in their paper, our method significantly outperformed their method, especially

at high levels of sensitivity. These results further support the claim that that SIDD is a better discriminator of promoter sequences than thermostability.

SIDD and -10 motifs together predict promoter and non-promoter sequences with high accuracy. As was demonstrated above, several types of DNA structural variables, as well as -10 motif scores, can differentiate promoters from non-promoter sequences. It should be possible to combine together those variables that are not highly correlated, to better discriminate promoters from non-promoter sequences. For each pair of attributes we have calculated their correlation over the entire training set. Table 2 shows the results of this analysis. We see that SIDD is not significantly correlated with curvature, moderately correlated with deformability and thermostability and moderately negatively correlated with -10 motif scores. Curvature is seen not to correlate with any other parameter used in this study.

When all these variables were combined in a linear discrimination function, the model can achieve more than 82% accuracy in predicting promoter and non-promoter sequences. However, when SIDD was combined only with -10 motif scores in a linear discrimination analysis, about 80% accuracy is achieved. This high accuracy was largely the result of a dramatic decrease in the false positive rate that occurred when SIDD properties were included. Similar results were obtained in a eukaryotic promoter prediction program McPromoter[20]. There a combination of sequence information with physical properties achieved a 30% reduction of false positives, when compared with the sequence model alone.

A procedure shown in Figure 5 was used to scan for possible promoter regions in the *E. coli* K12 genome. Without post-processing the results from linear discrimination function, 10069 potential promoter sequences ranging from 100bp to about 1700 bp were obtained for the whole genome, more than 73% of which were short than 200bp. 923 out of 927 documented TSS were found in this set. Multiple TSSs were observed to be located in a single predicted region, especially for those bigger than 200 bps. We expect these potential regions can be useful references for both bioinformatics and experimental research.

Table 2. Correlation coefficients between structural parameters, -10 motif scores in promoter sequences

	SIDDsum	Curvature	Deformability	Thermo-stability	- 10 motif scores
SIDDsum		-0.0847	0.5194	0.3009	-0.5652
SIDDmin	0.9145	-0.0811	0.4714	0.2856	-0.4990
Curvature	-0.0847		-0.1499	-0.0207	0.1873
Deformability	0.5194	-0.1499		0.5317	-0.7546
Thermo-stability	0.3009	-0.0207	0.5317		-0.4175
-10 motif scores	-0.5652	0.1873	-0.7546	-0.4175	

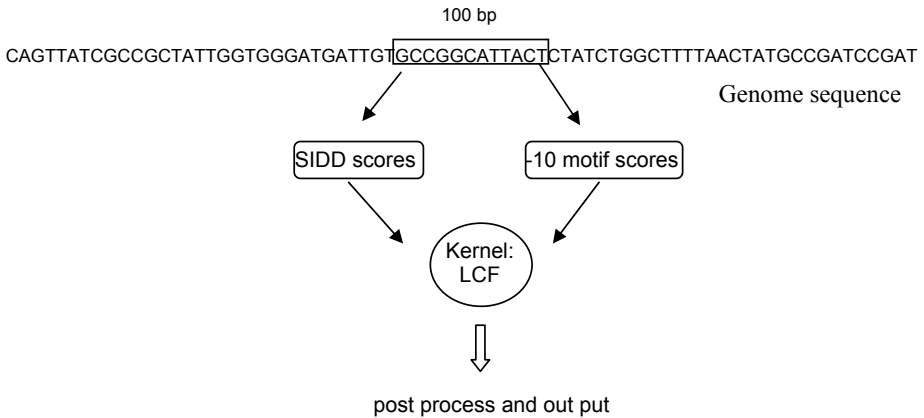


Fig. 5. Overview of a promoter region prediction program combining SIDD and -10 motif scores

3 Conclusions

In this report we show the propensity to undergo stress-induced duplex destabilization (SIDD) is a distinctive structural property of prokaryotic promoter sequences that is not directly related to primary sequence alone. Comparisons with other important structural properties, and with conserved -10 motifs, show SIDD to be the best discriminator between promoter and non-promoter sequences. We have developed methods using SIDD either as a sole predictor or in combination with other DNA structural and sequence properties to identify promoter sequences in test sets from prokaryotic genomes. The inclusion of SIDD properties is shown to greatly reduce the false positive prediction rate, while increasing the true positive rate. When applied to the two best experimentally annotated bacterial genomes from different phyla and having different nucleotide compositions, this approach achieved comparable levels of accuracy in both. Based on the distribution patterns of SIDD sites found in all prokaryotic genomes analyzed to date, we anticipate that SIDD-based methods could be useful in annotating promoter regions in complete prokaryotic genomes.

4 Methods

Sequences. We analyzed version M54 of the *E. coli* K12 genome, containing 4639221 base pairs. 927 experimental characterized transcriptional start sites (TSS) were obtained from the Regulon database [21]. The *Bacillus subtilis* genome analyzed is the version revised on 29-July-2004, containing 4214630 base pairs. 480 experimentally characterized TSSs were obtained from DBTBS[22]. In our analysis the promoter regions were represented by the 100 base pair DNA fragments from position -79 to +20 relative to the TSSs. The promoter training set for *E. coli* consisted of 500 such randomly chosen promoter regions; the other 427 promoters were used as our test set. The promoter training set of *Bacillus subtilis* consisted of

250 randomly chosen documented promoter regions. The rest of the 230 promoter regions were used as test set. The set of coding regions were selected as 100 base pair DNA fragments starting from position +300 relative to (i.e. downstream from) the TSSs. The CON regions were chosen as 100 base pair DNA fragments in the middle of correspondent convergent intergenic regions. The coding and CON data sets each consisted of 500 regions.

Curvature calculations. The predicted values of DNA curvature were calculated using the CURVATURE program [23]. This program was used to create a curvature map of the entire genome. For each base pair in the genome, the curvature value (in curvature units, cu) corresponds to the curvature of the calculated path of a 121 bp segment centered at that base pair.

Protein-induced deformability calculations. Values of the local protein-induced deformability were calculated using the dinucleotide model developed by Olson et al [24]. For each base pair in the genome, the deformability value is calculated as the average of the conformational volumes covered by its two neighboring DNA dimers in protein-DNA complexes.

Thermostability calculations. The thermodynamic stability profiles were calculated using the nearest-neighbor (NN) thermodynamics presented by SantaLucia et al. [25]. For each base pair in the genome, the value for its thermostability is calculated as the average of the opening energies for the two dinucleotides that contain it.

SIDD profile calculations. The predicted values of the destabilization energy $G(x)$ were calculated using the method of Benham and Bi [11]. The destabilization free energy associated to each base pair is the incremental free energy needed to guarantee its opening under the assumed superhelicity.

DNA sequence conservation and position specific score matrix (PSSM) calculation. The conservation of multiple aligned sequences can be evaluated using the Shannon entropy of information theory [26]. Sequence motifs or conserved sequences are here evaluated using position-specific scoring matrices (PSSMs) calculated using the method of Durbin [3].

Probability density estimation and statistical tests. The distributions of DNA structural properties or -10 motif scores in promoter regions, coding regions and CON regions were represented by their probability densities. The density for each property and type of region was evaluated using a normal kernel smoother at 100 equally spaced points covering the range of the data. The comparison of two distributions was made using the Kolmogorov-Smirnov two-sample test [27]. The null hypothesis for this test is that the two compared datasets are from the same continuous distribution.

Linear discrimination analysis. We combined DNA structural properties (destabilization free energy, curvature, or thermostability) with -10 sequence motif scores into a linear discrimination model [28]. Because promoter identification is a two-class classification, it is implemented using Fisher linear discriminant analysis. The procedure of linear discriminant analysis is to find a linear combination of the measures that provides maximum discrimination, in out case between promoter and

non-promoters. It assumes that the training sets are normally distributed. The scores of a data point D can be calculated as the dot product of $x \cdot w$, where $w = S^{-1} * (\mu_1 - \mu_0)$; where S is the pooled covariance matrix of the parameters, μ_1 and μ_0 are the sample mean vectors of parameters for the positive and negative data respectively. The vector w maximizes the ratio of inter-class variation of score to intra-class variation of score. A data point is classified into class 1 if the score is $D > c$, or into class 0 if the score is $D < c$; where $c = w * (\mu_1 - \mu_0) / 2$.

The true positive rate, false positive rate and accuracy are defined as follows (where TP = true positives, FP = false positives, TN = true negatives and FN = false negatives):

$$\text{True positive rate} = \frac{TP}{TP + FN}; \text{ False positive rate} = \frac{FP}{TN + FP}; \text{ Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

Acknowledgment. The work reported here was supported in part by grants to CJB from the National Science Foundation (DBI 0416764) and the National Institutes of Health (RO1-GM68903).

References

- Hayes, W.S. and M. Borodovsky, *How to interpret an anonymous bacterial genome: machine learning approach to gene identification*. Genome Res, 1998. **8**(11): p. 1154-71.
- Delcher, A.L., et al., *Improved microbial gene identification with GLIMMER*. Nucleic Acids Res, 1999. **27**(23): p. 4636-41.
- Durbin, R., et al., *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. 1998, Cambridge, UK: Cambridge University Press.
- Harley, C.B. and R.P. Reynolds, *Analysis of E. coli promoter sequences*. Nucleic Acids Res, 1987. **15**(5): p. 2343-61.
- Hawley, D.K. and W.R. McClure, *Compilation and analysis of Escherichia coli promoter DNA sequences*. Nucleic Acids Res, 1983. **11**(8): p. 2237-55.
- Rosenberg, M. and D. Court, *Regulatory sequences involved in the promotion and termination of RNA transcription*. Annu Rev Genet, 1979. **13**: p. 319-53.
- Helmann, J.D. and P.L. deHaseth, *Protein-nucleic acid interactions during open complex formation investigated by systematic alteration of the protein and DNA binding partners*. Biochemistry, 1999. **38**(19): p. 5959-67.
- Huerta, A.M. and J. Collado-Vides, *Sigma70 promoters in Escherichia coli: specific transcription in dense regions of overlapping promoter-like signals*. J Mol Biol, 2003. **333**(2): p. 261-78.
- Hertz, G.Z. and G.D. Stormo, *Escherichia coli promoter sequences: analysis and prediction*. Methods Enzymol, 1996. **273**: p. 30-42.
- Benham, C.J., *Sites of predicted stress-induced DNA duplex destabilization occur preferentially at regulatory loci*. Proc Natl Acad Sci U S A, 1993. **90**(7): p. 2999-3003.
- Benham, C.J. and C. Bi, *The analysis of stress-induced duplex destabilization in long genomic DNA sequences*. J Comput Biol, 2004. **11**(4): p. 519-43.

12. Wang, H., M. Noordewier, and C.J. Benham, *Stress-induced DNA duplex destabilization (SIDD) in the E. coli genome: SIDD sites are closely associated with promoters*. *Genome Res*, 2004. **14**(8): p. 1575-84.
13. Wang, H., M. Kaloper, and C.J. Benham, *SIDDBASE: A Database Containing the Stress-Induced DNA Duplex Destabilization (SIDD) Profiles of Complete Microbial Genomes*. *Nucleic Acids Res*, 2006. **34**: p. D1-D6.
14. Kanhere, A. and M. Bansal, *A novel method for prokaryotic promoter prediction based on DNA stability*. *BMC Bioinformatics*, 2005. **6**(1): p. 1.
15. Sheridan, S.D., C.J. Benham, and G.W. Hatfield, *Activation of gene expression by a novel DNA structural transmission mechanism that requires supercoiling-induced DNA duplex destabilization in an upstream activating sequence*. *J Biol Chem*, 1998. **273**(33): p. 21298-308.
16. Kozobay-Avraham, L., S. Hosid, and A. Bolshoy, *Curvature distribution in prokaryotic genomes*. *In Silico Biol*, 2004. **4**(3): p. 361-75.
17. Pedersen, A.G., et al., *A DNA structural atlas for Escherichia coli*. *J Mol Biol*, 2000. **299**(4): p. 907-30.
18. Lisser, S. and H. Margalit, *Determination of common structural features in Escherichia coli promoters by computer analysis*. *Eur J Biochem*, 1994. **223**(3): p. 823-30.
19. Reese, M.G., *Application of a time-delay neural network to promoter annotation in the Drosophila melanogaster genome*. *Comput Chem*, 2001. **26**(1): p. 51-6.
20. Ohler, U., et al., *Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition*. *Bioinformatics*, 2001. **17 Suppl 1**: p. S199-206.
21. Salgado, H., et al., *RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12*. *Nucleic Acids Res*, 2004. **32**(Database issue): p. D303-6.
22. Makita, Y., et al., *DBTBS: database of transcriptional regulation in Bacillus subtilis and its contribution to comparative genomics*. *Nucleic Acids Res*, 2004. **32**(Database issue): p. D75-7.
23. Shpigelman, E.S., E.N. Trifonov, and A. Bolshoy, *CURVATURE: software for the analysis of curved DNA*. *Comput Appl Biosci*, 1993. **9**(4): p. 435-40.
24. Olson, W.K., et al., *DNA sequence-dependent deformability deduced from protein-DNA crystal complexes*. *Proc Natl Acad Sci U S A*, 1998. **95**(19): p. 11163-8.
25. SantaLucia, J., Jr., *A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics*. *Proc Natl Acad Sci U S A*, 1998. **95**(4): p. 1460-5.
26. Shannon, C.E. and W. Weaver, *The Mathematical Theory of Communication*. 1949, Urbana, IL: The University of Illinois Press.
27. Sokal, R.R. and F.J. Rohlf, *Biometry*. Third Edition ed. 1995, New York: W.H. Freeman and Company.
28. Johnson, R.A. and D.W. Wichern, *Applied Multivariate Statistical Analysis*. Fifth Edition ed. 2002, Upper Saddle River, N. J.: Prentice Hall.

An Interaction-Dependent Model for Transcription Factor Binding

Li-San Wang¹, Shane T. Jensen², and Sridhar Hannenhalli^{3,*}

¹ Department of Biology

lswang@mail.med.upenn.edu

² Department of Statistics, The Wharton School

stjensen@wharton.upenn.edu

³ Penn Center for Bioinformatics and Department of Genetics

sridharh@pcbi.upenn.edu

University of Pennsylvania

Philadelphia, PA 19104-6021

Abstract. Transcriptional regulation is accomplished by several transcription factor proteins that bind to specific DNA elements in the relative vicinity of the gene, and interact with each other and with Polymerase enzyme. Thus the determination of transcription factor-DNA binding is an important step toward understanding transcriptional regulation. An effective way to experimentally determine the genomic regions bound by a transcription factor is by a ChIP-on-chip assay. Then, given the putative genomic regions, computational motif finding algorithms are applied to estimate the DNA binding motif or positional weight matrix for the TF. The *a priori* expectation is that the presence or absence of the estimated motif in a promoter should be a good indicator of the binding of the TF to that promoter. This association between the presence of the transcription factor motif and its binding is however weak in a majority of cases where the whole genome ChIP experiments have been performed. One possible reason for this is that the DNA binding of a particular transcription factor depends not only on its own motif, but also on synergistic or antagonistic action of neighboring motifs for other transcription factors. We believe that modeling this interaction-dependent binding with linear regression can better explain the observed binding data. We assess this hypothesis based on the whole genome ChIP-on-chip data for Yeast. The derived interactions are largely consistent with previous results that combine ChIP-on-chip data with expression data. We additionally apply our method to determine interacting partners for CREB and validate our findings based on published experimental results.

1 Introduction

Gene transcription in eukaryotes is tightly regulated, both spatially and temporally, by several transcription factors (TF) [1, 2]. Thus, a first step in computational analysis of transcription is to model the DNA binding preference of the TFs, which can be done using a set of experimentally determined DNA sequences bound by a TF. Traditionally this was either done using *in vitro* SELEX experiment [3] or

* Corresponding author.

footprinting assay [4] to determine binding sites followed by specific mutagenesis of the binding site to determine the acceptable bases. Either of these technologies result in short sequences, which can be aligned to derive the binding specificity as a *positional weight matrix (PWM)* [5]. A recent high throughput technique to determine the genomic regions bound by a specific transcription factor is *Chromatin Immunoprecipitation (ChIP)* experiments [6]. Although these regions are large (few hundred bases), computational motif discovery algorithms can be applied to the bound regions in order to detect the most likely PWM for the TF [5].

Critical to transcriptional regulation, is the interaction among the TFs, which in turn depends on TFs binding to specific *cis* elements in relative vicinity. This interaction-dependent functionality is the basis for *transcriptional modules* [7-9]. In some cases, this dependency is involved in protein modifications, like phosphorylation. However, there are other cases where the binding of the TF itself is interaction-dependent [10, 11]. This interaction-dependent binding often results in inconsistency between binding specificity of a factor and experimentally determined binding sites for the factor. For example, transcription factor CREB is believed to bind to the cAMP response element (CRE), which has motif CGTCA. However, based on genome wide Chromatin Immunoprecipitation (ChIP) assay, 70% of CREB-bound regions do not contain the CGTCA motif in the ~800 bps region [12]. More generally, we have found that for a majority of factors for which genome-wide ChIP-chip assay has been performed, the derived PWM does not sufficiently explain the experimental binding data.

We investigate whether, modelling the binding of a TF as an interaction-dependent mechanism can lead to better predictions of the binding of a TF to a promoter region. We model the binding probability of a TF as a linear combination of PWM scores corresponding to that TF as well as other potentially interacting TF. Similar linear model was used to model the regulation of a gene by the promoter motifs in [13]. We optimize the coefficients using linear regression. The estimated coefficients of this linear combination indicate the degree of dependence between the TFs and the sign of the coefficients indicate whether the interaction is synergistic or antagonistic. Our interaction-dependent binding model better estimates the experimentally determined binding probabilities and also reveals specific synergistic and antagonistic interactions among factors in yeast. We compare these detected interactions with previous methods [9, 14] and discuss the novel ones. We further apply our method to rat genomic regions bound by CREB to determine its interacting factors. Most of the detected CREB partners are validated using published experimental literature.

2 Results

We use the ChIP-chip data for 204 transcription factors on 6229 genes in yeast [15]. Of these 204 TFs, motif or a PWM derived from the bound promoters is provided for 102 TFs. We evaluate our method on 90 of these transcription factors (see methods). Using these motifs, we estimate the dependence of a factor on other factors for its binding to a promoter. The dependence coefficients are computed using linear regression so as to minimize the overall difference between the experimental binding probability and estimated binding probability. We use the average sum of squared errors (SSE – see methods) to compare the relative

advantage of using interaction-dependent model in estimating the binding probabilities. For each TF, we consider r genes for which the binding p-value is less than 0.001, as suggested in [15]. We also include r genes with the largest p-values as well as $2r$ randomly selected genes from among the remaining genes. This set of $4r$ genes is split into *training*, *model selection*, and *test* sets described later.

Improved estimate of binding probability in yeast using pair-wise interactions. The best pair-wise interaction regression model is created for each of the 58 TFs where the number of binding genes is at least 20 ($r = 20$), so that the regression model has at least 80 observations (r bound, r unbound and $2r$ random genes; see methods). For several TFs, a model without any interaction terms (the *interaction-free* model) leads to small prediction errors on its own. These TFs may bind in an interaction-independent fashion and thus do not require a more detailed model. Thus, if we consider only the TFs which have “high” sum of squared errors (SSE) assuming interaction-independent binding, we would expect a greater fraction of these TFs to have an improved SSE when interaction is used. As shown in Fig. 1, the binding data for many TFs is better modelled by including interactions between TFs and the fraction of improved TFs increases with the SSE of interaction-independent binding. As a negative control we calculated the interaction-dependent SSE for a randomly chosen interacting TF (shown by the striped bars in Fig. 1). The fractions of TFs that are helped by including interactions is much greater than the fraction when we randomly select the interacting partner instead of the best possible interacting partner (based on our model selection).

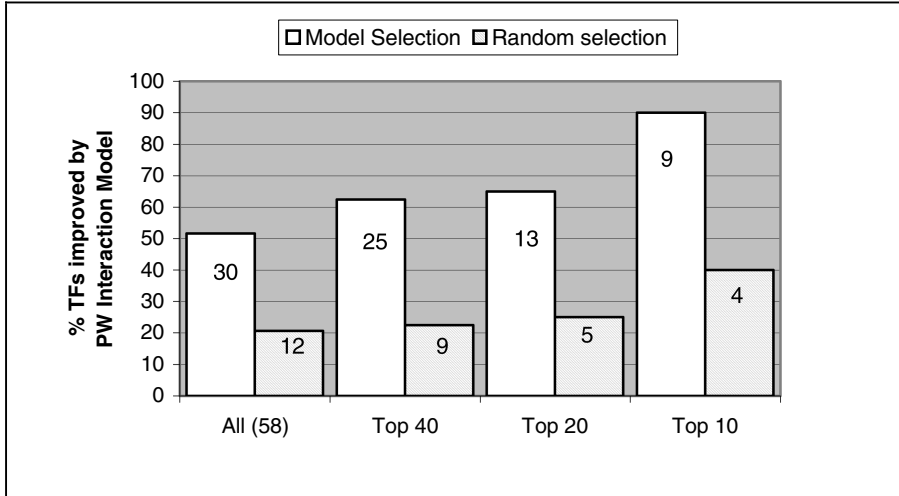


Fig. 1. Percent improved in bins using pair-wise interactions. Using the top X ($X=58, 40, 20, 10$) of all 58 TFs sorted in decreasing order of their SSE with non-interacting models, we calculate the fraction of TFs where the SSE is smaller using the pair-wise interaction model than that for the non-interaction model. We use either model selection or random selection as a control to determine which pair of TFs interact (see methods section). On each bar is the number of TFs improved.

Using three-way interactions. We also investigated whether the model for some TFs can be additionally improved by extending the number of allowed interactions and the results are shown in Fig. 2. The fraction of TFs for which the three-way interaction model is a better fit to the experimental data is reduced compared with the pair-wise case. It is however, still significantly greater relative to the case when the interacting partners are selected randomly.

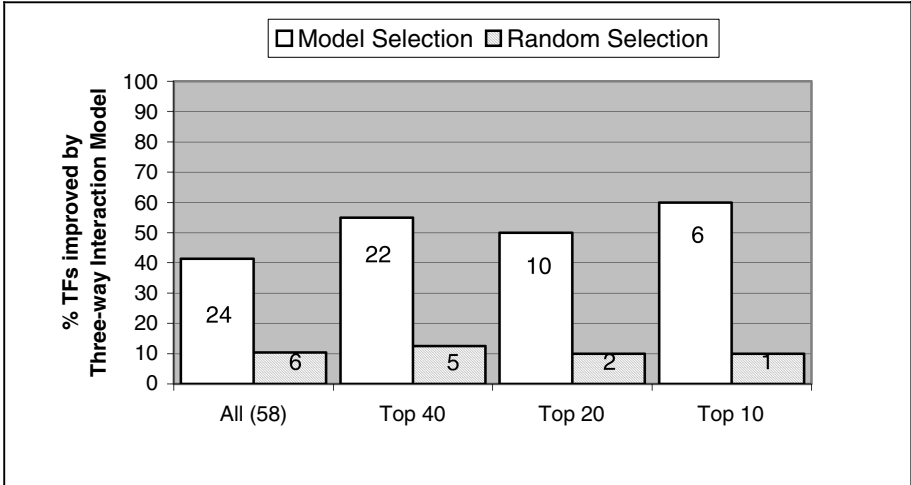


Fig. 2. Percent improved in bins using three-way interactions. Using the top X ($X=58, 40, 20, 10$) of all 58 TFs sorted in decreasing order of their SSE with non-interacting models, we calculate the fraction of TFs where the SSE is smaller using the pair-wise interaction model than that for the non-interaction model. We use either model selection or random selection as a control to determine which pair of TFs interact (see methods section). On each bar is the number of TFs improved.

Identification of interactions using entire data. The interaction between transcription factors are represented in our model by regression coefficients. Positive coefficients imply *synergistic* interactions and the negative coefficients imply *antagonistic* interactions. By applying a FDR threshold of 0.1 to these coefficients, we obtained 377 synergistic interactions and 68 antagonistic interactions.

Yeast cell cycle. We compared our finding with the combinations of TFs regulating the stages of cell cycle that was reported in [9], based on a combination of ChIP-on-chip and conditional expression data. Excluding SKN4, which was excluded from our analysis because of a small number of gene it bind to, there are 18 known interactions among 11 factors. Using our 377 predicted synergistic interactions, which represents 9.4% of all possible interactions among 90 TFs used in our analysis, we predict 34 (62%) of the possible 55 interactions among TFs in the cell cycle, and 16 (89%) of the 18 known interactions. The p-values of our interaction coefficients are significantly lower for the 18 known interactions compared with the other pairs of cell cycle TFs. Our approach accurately identifies (with ~89% sensitivity) the known interactions in

cell cycle regulation at a low overall prediction rate of 9.4%. Cell cycle factors were not included in any of the 68 antagonistic interactions predicted by our method.

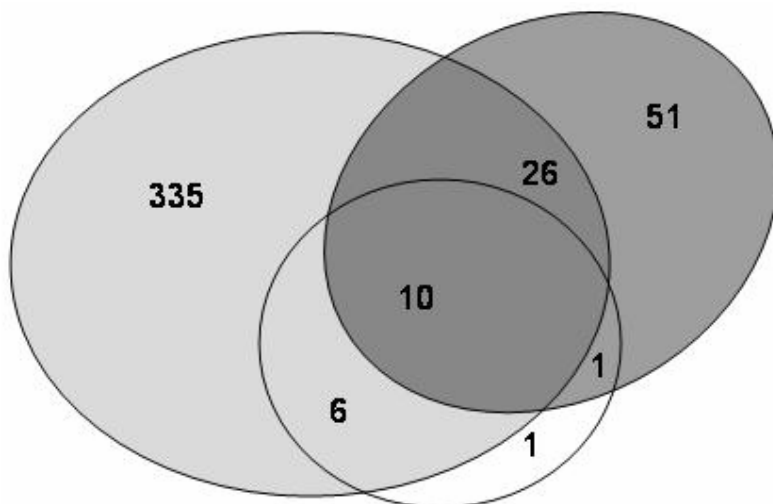


Fig. 3. Comparing three sets of interactions. White ellipse: 18 cell cycle interactions, gray ellipse: 377 regression based predictions, Dark ellipse: 98 interactions from [14].

Comparison with other previously predicted interactions. Banerjee et al. used *expression coherence* to predict cooperativity between transcription factors [14]. The hypothesis being: if the genes targeted (based to ChIP data) both by factors x and y have more similar expression profiles relative to the genes targeted by either x or y , but not both, then x and y are likely to cooperate. The same idea was earlier used in [16]. We took the 193 predicted pairs at p-value of 0.01 in [14]. Filtering out the TFs not represented in our set of 90 factors results in 98 predicted pairs. Fig. 3 shows the comparison of three sets, (1) Our 377 predicted synergistic interactions, (2) 98 interactions from [14], and (3) 18 cell cycle interactions from [9]. Banerjee et al. identify 11 (61%) of the cell cycle interactions, compared with our 89%, while predicting far fewer interaction (98 versus 377). However it should be noted that they used both the ChIP data and the expression data as did Lee et al. in inferring the cell cycle interactions while we have relied solely on ChIP data. Clearly, it is desirable to use additional, functional and expression information, to predict TF-TF interactions. Our aim however, is to predict the dependencies strictly in TF-DNA binding. TF-DNA binding may not correspond to expression of the target gene since it depends on post-translational modification of the transcription factors. Our result, based solely on the ChIP data, is largely consistent with previously known interactions and proposes several novel interactions.

Antagonism. An important aspect of our regression-modeling approach is that we detected 68 pairs of antagonistic interactions, in addition to our predicted synergistic interactions. Unfortunately, there is no experimental data on TF antagonism that we

could use to validate our findings, so we instead used the following approach to indirectly test our findings. If TF x antagonizes TF y , then for all the promoters which contain a motif for y , we should see an inverse correlation between the presence of motif x and the binding of TF y . In other words, the presence of motif x should repress the binding of TF y . For each such pair (x y), we first selected the top 100 genes with strongest motif hits for TF y . For these 100 genes we computed the Pearson's correlation coefficient between 100 occupancy probabilities for TF x and the 100 ChIP binding values for TF y . We computed the correlation for the antagonistic pairs and 500 randomly selected pairs. The average correlation coefficient for the random pairs is 0.004, whereas that for the antagonistic pairs is -0.05. Although this difference is small, it is significant with a p-value of 0.0003 based on t-test and with a p-value of 0.003 based on Kolmogorov-Smirnov test.

Detection of CREB binding partners. As the positive set for CREB bound regions we extracted the ± 500 bp regions flanking the 5948 high-confidence CREB bound locations in rat based on SACO technology [17]. We additionally extracted 5000 random 1 kb regions from the rat genome. Similarly to Yeast analysis, we computationally determined the occupancy probability for each of the 546 vertebrate PWMs from TRANSFAC 8.4 against each of these 10948 sequences. Unlike the Yeast ChIP-chip data, in this case, we do not have an experimental binding p-value for CREB against these sequences. Instead we use a value of 1 for the positive sequences and a value of 0 for the background sequences. Based on the pair-wise regression, using FDR threshold of 0.05, ie. 5% false discovery rate, we discovered 18 TFs that synergistically interact with CREB. Using the published literature, 10 of these have evidence for interaction with CREB, majority of which are via the CREB co-activator CBP. These are TBP [18], STAT1 [19], GR [20], GATA-1 [21], p53 [22], Sp-1 [23], AhR (via Arnt and CBP) [24], v-Myb [25], SMAD-3 [26], and Major_T-antigen [27]. Additionally MEIS1 has a CBP-dependent Protein-Kinase-A response domain [28]. For additional 4 factors, we could not find direct evidence in the literature. However the DNA binding specificities for these factors are very similar (details excluded) to another factor for which there was evidence for interaction with CREB. In terms of DNA binding specificity, Ncx is similar to STAT1, GEN_INI to CREB, and MAZR to Sp-1, and AP-4 to MyoD. For 3 of the 18 factors – Pax, ZF5, and XPF-1, we could not find any literature evidence of interaction with CREB.

3 Methods

Data. We use the genome scale ChIP data in Yeast for 204 transcription factors for 6229 genes reported in [15]. The binding data is expressed as a p-value for the null hypothesis of no binding. The lower the p-value for a gene, the more confident we are that there is binding between the gene and the TF. By “gene” we implicitly mean the upstream 700 base pairs of the gene which was used to design the microarray in [15]. Using an extensive set of motif detection algorithms and a supervised integration of their results, the authors also provide the log-odds matrix or positional weight matrix (PWM) for 102 of these factors, some of which are identical (for example, MET31 and MET32). Using identical PWMs in our regression model would lead to identical

coefficients, so our analysis focused on a subset of 90 of these 102 transcription factors that have unique PWMs. For a factor t and gene g represent the binding probability as $ChIP(t,g)$. Accordingly we extracted these regions using the yeast genome sequence as well as the gene annotation from the SGD database at Stanford University (<ftp://genome-ftp.stanford.edu/>). The $ChIP(t,g)$ p-value was transformed to a larger and more continuous scale by the transformation $\log(0.001 / ChIP(t,g))$. This transformation has been previously shown to be effective in [29].

Occupancy probability of a PWM in a gene promoter. Positional Weight matrix (PWM) is a 4 by w matrix representing the DNA binding specificity of a TF that binds to a w bases long DNA string. For a PWM M , the entry M_{ib} at position i for base b is the $-\ln(P_{ib}/Q_b)$, where P_{ib} is the frequency of base b at position i among the known binding sites and Q_b is the background frequency of base b [5]. Given the PWM M and a sequence $s = (s_1, s_2, \dots, s_w)$, the raw score r of M on s is computed as

$r(M, s) = \sum_{i=1}^w M_{is_i}$, where M_{ib} is the entry in M at position i for base b . The transformed score $x(M, s) = e^{r(M, s)}$ is proportional to the binding energy of the TF to the sequence. The score of M on a 700 bp gene promoter p : $x(M, p) = \sum_{s \in P} x(M, s)$ where s represents all w -long substrings in p in either strand. Let $Z = \sum_{p \in P} x(M, p)$ where P represents the universe of “all”

promoters in the genome to which M has access to. The probability that M occupies a promoter p is estimated as $x(M, p)/Z$. However only a small fraction of all promoters are actually accessible to the TF for binding at any given time and we should use only the sum of scores over this subset for normalization. The true occupancy probability is hard to compute as it depends on several other attributes, like the state of the chromatin, concentration of the TF, binding affinity of the TF, genome composition etc. Thus the number we compute serves as a proxy for the occupancy probability.

Modeling the TF-GENE binding using linear regression and model selection. We use the following regression model:

$$Y_{ij} = \mu_j + a_j x_{ij} + \sum_{k \in R_j} b_{jk} x_{ik} + \varepsilon_{ij}$$

Here R_j is a set of TFs (excluding TF j itself) interacting with TF j that will be determined using an iterative model selection procedure detailed below, $Y_{ij} = \log(0.001/p_{ij})$, where p_{ij} is the ChIP-chip based p-value of binding for TF j and gene i , x_{ij} is the occupancy probability of TF j to the upstream region of gene i , μ_j , a_j , and b_{jk} are regression coefficients, and ε_{ij} is the normally distributed error term. We use ordinary least squares regression to solve for the coefficients and their significance.

For each TF, we consider r genes for which the binding p-value ≤ 0.001 , as suggested in [15]. We also include r genes with largest p-values (least likely to bind) as well as $2r$ randomly selected genes from among the remaining genes. We run validation tests 10 times for each TF. For each validation test, the set of $4r$ genes is randomly split into *training* (50%), *model selection* (25%), and *test sets* (25%). We run the following procedure to determine set R_j , the coefficients, and the error of the regression. Initially we set R_j to be empty. At each iteration we augment R_j by one

new TF as follows. For each TF k' not in R_j , set $R_{jk'} = R_j \cup \{k'\}$. We run the regression using $R_{jk'}$ on the training gene set to obtain the coefficients, and use the model selection gene set to compute the sum-of-squared-errors (SSE) of the coefficients. We find the TF k'' with the smallest SSE on the model selection genes, and add k'' to R_j . We limit the size of R_j to 2 to avoid over-fitting. At the end of the procedure, we compare the square-root of the average SSE (SSE divided by r , the number of test genes) of the newly obtained model with the non-interacting model (assuming R_j is empty) on the set of test genes to see whether the interacting model has smaller error. To assess whether the improvement is due to chance alone, we also incorporated random-selection as the null distribution: we follow all steps as above except when augmenting R_j ; instead of selecting the TF with the smallest error to add, we randomly select the TF.

Identification of interactions using entire data. The regression model assuming that TF k affects the binding behavior of TF j is $Y_{ij} = \mu_j + a_j x_{ij} + b_{jk} x_{ik} + \varepsilon_{ij}$ (the model is the same as that in previous section when $R_j = \{k\}$ for some $k \neq j$). The p-value for the assumed interaction is the significance of b_{jk} , a positive b_{jk} represents *synergistic* interaction, and a negative b_{jk} *antagonistic*. To account for multiple-testing, we estimated the false discovery rate (FDR)¹ corresponding to the p-values. Unless otherwise mentioned, we use a FDR threshold of 0.1. By applying these thresholds, we obtain 377 synergistic interactions and 68 antagonistic interactions.

4 Discussion

The importance of combinatorial interaction among transcription factors for proper transcriptional regulation is widely acknowledged. However, the DNA-binding of a factor itself is often treated as an isolated independent event, despite evidence to the contrary. We have shown that by considering dependence between TFs we can better explain the experimental TF-DNA binding data. Unlike the interactions discovered by various other methods [9, 14, 31] the proposed method predicts antagonistic interactions as well as the synergistic ones. Our method accurately identifies most of the TF-TF interactions of the yeast cell cycle. Although a similar validation for antagonistic interactions is currently not possible, but we have shown that the presence of antagonizing factor's motif reduces the binding probability for the antagonized factor. This inverse correlation is modest but significant.

Our attempts to include more than two additional TFs in the linear regression, does not generally yield an improved model. However, a natural extension of our method is to investigate the optimal number of interacting TFs. This can be done by incrementally adding additional TF in the regression that yields the maximum reduction in SSE. Based on our investigation, we do not expect a majority of factors

¹ FDR control procedure takes as input the p-values of all genes and FDR threshold (eg. 0.1), and finds a maximum p-value threshold for significance, such that the proportion of false positives out of all called positives (significant genes) never exceeds the given FDR threshold. We use the Bioconductor library for R to perform the FDR control, which implements the method described in [30].

to depend on multiple other factors for their binding, and for the subset which might depend on several other factors, perhaps a non-linear model would lead to better predictions. A lack of sufficient data (number of genes bound to a TF) as well as the inherent experimental noise make this a difficult proposition.

Similar to other computational analysis of gene regulation, there are several limitations to our approach. We implicitly assume that all transcription factors are available in the cell and the only determinant of binding is the interaction. We have essentially ignored the dynamic nature of binding owing to presence/absence of factors, co-factors, chromatin structure etc. The available ChIP data presents only a single snap-shot of the cell, and ChIP data under different conditions may reveal slightly different relationships among factors. This general limitation of the available data affects all computational approaches for predicting interactions. Finally, we have assumed a known motif for each factor which was computed based on the bound set of promoters, but an approach that attempts to simultaneously detect the motifs as well as the interactions should be more effective.

References

1. Ptashne, M. A genetic switch, Edn. third. (Cold Spring Harbor Laboratory Press, 2004).
2. Kadonaga, J.T. Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. *Cell* 116, 247-257 (2004).
3. Tuerk, C. & Gold, L. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* 249, 505-510 (1990).
4. Guille, M.J. & Kneale, G.G. Methods for the analysis of DNA-protein interactions. *Mol Biotechnol* 8, 35-52 (1997).
5. Stormo, G.D. DNA binding sites: representation and discovery. *Bioinformatics* 16, 16-23 (2000).
6. Horak, C.E. & Snyder, M. ChIP-chip: a genomic approach for identifying transcription factor binding sites. *Methods Enzymol* 350, 469-483 (2002).
7. Bolouri, H. & Davidson, E.H. Modeling DNA sequence-based cis-regulatory gene networks. *Dev Biol* 246, 2-13 (2002).
8. Thompson, W., Palumbo, M.J., Wasserman, W.W., Liu, J.S. & Lawrence, C.E. Decoding human regulatory circuits. *Genome Res* 14, 1967-1974 (2004).
9. Lee, T.I. et al. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* 298, 799-804 (2002).
10. Lomvardas, S. & Thanos, D. Nucleosome sliding via TBP DNA binding in vivo. *Cell* 106, 685-696 (2001).
11. Hochschild, A. & Ptashne, M. Cooperative binding of lambda repressors to sites separated by integral turns of the DNA helix. *Cell* 44, 681-687 (1986).
12. Euskirchen, G. et al. CREB binds to multiple loci on human chromosome 22. *Mol Cell Biol* 24, 3804-3814 (2004).
13. Bussemaker, H.J., Li, H. & Siggia, E.D. Regulatory element detection using correlation with expression. *Nat Genet* 27, 167-171. (2001).
14. Banerjee, N. & Zhang, M.Q. Identifying cooperativity among transcription factors controlling the cell cycle in yeast. *Nucleic Acids Res* 31, 7024-7031 (2003).
15. Harbison, C.T. et al. Transcriptional regulatory code of a eukaryotic genome. *Nature* 431, 99-104 (2004).

16. Pilpel, Y., Sudarsanam, P. & Church, G.M. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet* 29, 153-159 (2001).
17. Impey, S. et al. Defining the CREB regulon: a genome-wide analysis of transcription factor regulatory regions. *Cell* 119, 1041-1054 (2004).
18. Conkright, M.D. et al. Genome-wide analysis of CREB target genes reveals a core promoter requirement for cAMP responsiveness. *Mol Cell* 11, 1101-1108 (2003).
19. Valineva, T., Yang, J., Palovuori, R. & Silvennoinen, O. The transcriptional co-activator protein p100 recruits histone acetyltransferase activity to STAT6 and mediates interaction between the CREB-binding protein and STAT6. *J Biol Chem* 280, 14989-14996 (2005).
20. Imai, E., Miner, J.N., Mitchell, J.A., Yamamoto, K.R. & Granner, D.K. Glucocorticoid receptor-cAMP response element-binding protein interaction and the response of the phosphoenolpyruvate carboxykinase gene to glucocorticoids. *J Biol Chem* 268, 5353-5356 (1993).
21. Blobel, G.A., Nakajima, T., Eckner, R., Montminy, M. & Orkin, S.H. CREB-binding protein cooperates with transcription factor GATA-1 and is required for erythroid differentiation. *Proc Natl Acad Sci U S A* 95, 2061-2066 (1998).
22. Grossman, S.R. p300/CBP/p53 interaction and regulation of the p53 response. *Eur J Biochem* 268, 2773-2778 (2001).
23. Raychowdhury, R. et al. Interaction of early growth response protein 1 (Egr-1), specificity protein 1 (Sp1), and cyclic adenosine 3'5'-monophosphate response element binding protein (CREB) at a proximal response element is critical for gastrin-dependent activation of the chromogranin A promoter. *Mol Endocrinol* 16, 2802-2818 (2002).
24. Kobayashi, A., Numayama-Tsuruta, K., Sogawa, K. & Fujii-Kuriyama, Y. CBP/p300 functions as a possible transcriptional coactivator of Ah receptor nuclear translocator (Arnt). *J Biochem (Tokyo)* 122, 703-710 (1997).
25. Oelgeschlager, M., Janknecht, R., Krieg, J., Schreck, S. & Luscher, B. Interaction of the co-activator CBP with Myb proteins: effects on Myb-specific transactivation and on the cooperativity with NF-M. *Embo J* 15, 2771-2780 (1996).
26. Pouponnot, C., Jayaraman, L. & Massague, J. Physical and functional interaction of SMADs and p300/CBP. *J Biol Chem* 273, 22865-22868 (1998).
27. Love, T.M. et al. Activation of CREB/ATF sites by polyomavirus large T antigen. *J Virol* 79, 4180-4190 (2005).
28. Huang, H. et al. MEIS C termini harbor transcriptional activation domains that respond to cell signaling. *J Biol Chem* 280, 10119-10127 (2005).
29. Smith, A.D., Sumazin, P., Das, D. & Zhang, M.Q. Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics* 21 Suppl 1, i403-i412 (2005).
30. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. B* 85, 289-300 (1995).
31. Hannenhalli, S. & Levy, S. Predicting transcription factor synergism. *Nucleic Acids Res* 30, 4278-4284 (2002).

Computational Characterization and Identification of Core Promoters of MicroRNA Genes in *C. elegans*, *H. sapiens* and *A. thaliana*

Xuefeng Zhou^{1,*}, Jianhua Ruan^{1,*}, and Guandong Wang¹,
and Weixiong Zhang^{1,2,**}

¹ Department of Computer Science and Engineering,

² Department of Genetics

Washington University in Saint Louis

Saint Louis, MO 63130-4899, USA

{xzhou, jruan, gw2, lzhang}@cse.wustl.edu

Abstract. MicroRNAs are short, noncoding RNAs that play important roles in post-transcriptional regulation. Although many functions of microRNAs in plants and animals have been revealed in recent years, the transcriptional mechanism of microRNA genes is not well understood. To elucidate the transcriptional regulation of microRNA genes, we study and characterize, in a genome scale, the promoters of intergenic microRNA genes in *Caenorhabditis elegans*, *Homo sapiens* and *Arabidopsis thaliana*. Specifically, we show that the known microRNA genes in these species have the same type of promoters as the protein-coding genes. To further characterize the promoters of miRNA genes, we develop a miRNA core promoter prediction method, called *common query voting* (CoVote). We applied this new method to identify putative core promoters of most known microRNA genes in the three model species of choice.

1 Introduction

MicroRNAs (miRNAs) are endogenous single-stranded RNAs ranging from 19 to 25nt in length. They are generated from long precursors that fold into hairpin structures, and are known to repress post-transcriptional gene expression in both animals and plants [13]. The two best understood miRNAs, *lin-4* and *let-7*, were discovered in 1990s', and proved to regulate developmental timing in *C. elegans* by repressing the translation of a family of key mRNAs [9,15,22]. Since then, several hundred miRNAs have been identified in viruses, plants and animals, and their important post-transcriptional regulatory functions have been discovered.

The biogenesis of miRNAs is complex. Although some are originated from intronic regions, most miRNAs are encoded in their own genes situated in intergenic regions or located on the antisense strands of annotated genes [6,7,8]. The intergenic miRNA genes are believed to be transcribed independently and

* These authors contributed equally to this research.

** Corresponding author. Phone: (314)935-8788, Fax: (314)935-7302.

form a new gene family, whereas the intronic ones are transcribed with their host genes [11]. Our knowledge of post-transcriptional processing of miRNAs has greatly expanded in recent years through various studies [10,2,13]. However, our understanding of the transcription of miRNA genes, which is the first and an important step of miRNA biogenesis, is limited.

Although many pieces of evidence implied that miRNA genes are class-II genes, no direct evidence was provided until late 2004. The first direct experiment on a single miRNA gene was reported by Lee *et al.* in late 2004, showing that it can be transcribed by pol II [12]. They determined the promoter and terminator region of this gene. However, their results, especially those on the promoter of *mir-23a~27a~24-2*, do not match very well with our current knowledge of pol-II promoters. Specifically, the promoter of *mir-23a~27a~24-2* appears to lack the known common promoter elements required to initiate transcription, such as the TATA-box, initiator element, downstream promoter element (DPE), TFIIB recognition element (BRE) [12], or even the proximal sequence element (PSE). Additionally, they also found that a large proportion of a given pri-miRNA does not contain a 5' cap or a poly(A) tail [12]. Another experimental evidence is from a *M. musculus* miRNA gene, *mmu-mir-290~291~292~293~294~295*. Houbaviy *et al.* found a canonical TATA-box located at -35 of capped and polyadenylated primary transcript (pri-miRNA) of this gene, and showed that this upstream region was also conserved in the *H. sapiens* homologous gene, *hsa-mir-371~372~373* [4]. Furthermore, Xie *et al.* recently identified the promoters of 40 *A. thaliana* miRNA genes, and showed that most of them have TATA-boxes in their promoters [23].

All these results are fundamentally important; they provided direct evidence that a miRNA gene can be transcribed by pol II. However, a few critical questions remain yet to be answered. The first is whether *all* known miRNA genes of different species are class-II genes. Although more than 40 *A. thaliana* miRNA genes have been shown to be transcribed by pol II, evidence for transcription of miRNA genes in animals is still limited. The second question is where the core promoters of miRNA genes locate. The knowledge of the locations and structure of the miRNA gene core promoters will be very useful for further investigation on the transcriptional mechanism of miRNA genes.

We consider these important issues in this research through a genome-wide computational analysis on the known miRNA genes of three model species, *C. elegans*, *H. sapiens* and *A. thaliana*. Our overall strategy is based on the following perspective on transcriptional regulation. The class-II and class-III genes (genes transcribed by pol III) must have distinctive features in their promoter regions, including transcription factor binding motifs, to recruit the right transcriptional machinery to initiate their transcription. Based on this perspective and supported in part by the results in [4,12,23], we first assume that the promoters of intergenic miRNA genes share common sequence features with the promoters of known class-II or class-III genes. We then build computational models to separate the promoters of class-II and class-III genes as well as random sequences. Using these models, we test all the experimentally verified intergenic miRNA

genes in the three species to determine what type of promoters miRNA genes may have, and subsequently answer the question which RNA polymerase is responsible for the transcription of these miRNA genes. One way to answer the second question is to apply a promoter prediction method to predict the core promoters of miRNA genes first, and then verify the predictions by wet-lab experiments. However, all existing promoter prediction methods are not suitable for the miRNA genes since they were not trained by the promoters of miRNA genes. Unfortunately, the promoters of most miRNA genes remain undefined, and the miRNA genes in Arabidopsis studied in [23] are not sufficient to build a good predictive model even for Arabidopsis. To locate core promoters of miRNA genes, we propose a novel promoter prediction method, called *common query voting* (CoVote). Using our new promoter prediction method, we further investigate core promoter regions of miRNA genes in the three model species, *C. elegans*, *H. sapiens* and *A. thaliana*.

2 Results

2.1 Discriminative Models of Pol-II and Pol-III Promoters

It is believed that RNA polymerase II (pol II) and polymerase III (pol III) transcribe different types of genes whose promoters are intrinsically different from each other and other genomic sequences [18]. Therefore, it is viable to assume that the core promoters of these two classes of genes had discriminative sequence features which separate them from each other and distinguish them from the other genomic sequences. Consequently, a discriminative model can be built using the known promoters of these two types of genes, and used to characterize a query sequence, determining if it belongs to pol-II or pol-III promoters or other intergenic sequences.

Specifically, we built a three-class discriminative model, or classifier, to distinguish pol-II promoters, pol-III promoters, and random intergenic sequences for each of the three species, *C. elegans*, *H. sapiens* and *A. thaliana*. We first prepared three sets of training data, respectively, for pol-II promoters, pol-III promoters and random intergenic sequences for each of the three species (see Methods). The numbers of the promoter sequences for these species, each of which is 250bp long, are listed in Supplementary Table 1. We then extracted statistically over-represented sequence motifs of 5bp to 10bp long, from each training set separately using our WordSpy motif-finding algorithm [20] (see Methods). We used these motifs as features to develop discriminative models. In our study, we built and compared two types of discriminative models, one in support vector machines (SVM) [16] and the other in decision trees [14] (see Methods). We adopted these two well studied classification methods to ensure that our analysis of miRNA genes is not skewed by the computational methods used.

We evaluated the quality of the discriminative models in terms of sensitivity, specificity and accuracy. Table I lists the ten-fold cross-validation results of the SVM and decision-tree based classifiers. The results show that these discriminative models are fairly accurate, with the minimum accuracy greater than 96%

Table 1. Results of ten-fold cross validations of SVM and decision-tree models. (a): sensitivity = # correct predicted positives / # total positives. (b): specificity = # correct predicted positives / # total predicted positives. (c): accuracy = # correct predicted instances / # total instances.

SVM model					
species	Pol II		Pol III		overall accuracy ^(c)
	sensitivity ^(a)	specificity ^(b)	sensitivity ^(a)	specificity ^(b)	
<i>C. elegans</i>	0.987	0.993	0.968	0.994	0.989
<i>H. sapiens</i>	0.970	0.987	0.940	0.998	0.971
<i>A. thaliana</i>	0.836	0.985	0.971	0.998	0.964

Decision-tree model					
species	Pol II		Pol III		overall accuracy ^(c)
	sensitivity ^(a)	specificity ^(b)	sensitivity ^(a)	specificity ^(b)	
<i>C. elegans</i>	0.955	0.941	0.937	0.942	0.945
<i>H. sapiens</i>	0.909	0.897	0.900	0.922	0.874
<i>A. thaliana</i>	0.889	0.928	0.972	0.974	0.958

Table 2. Classification results of promoter sequences of protein coding genes. (a): At least one segment was classified as pol-II promoter, and all other segments were classified as random intergenic sequences. (b): More segments were classified as pol-II promoters than pol-III promoters. (c): More segments were classified as pol-III promoters than pol-II promoters. (d): At least one segment was classified as pol-III promoter, and all other segments were classified as random intergenic sequences. (e): All segments were classified as random intergenic sequences.

promoter class	<i>C. elegans</i>	<i>H. sapiens</i>	<i>A. thaliana</i>
Pol-II only ^(a)	921 (92.1%)	630 (63.0%)	693 (69.3%)
Pol-II > pol-III ^(b)	8 (0.8%)	343 (34.3%)	20 (2%)
Pol-II ≤ pol-III ^(c)	6 (0.6%)	24 (2.4%)	20 (2%)
Pol-III only ^(d)	2 (0.2%)	1 (0.1%)	11 (1.1%)
Random only ^(e)	63 (6.3%)	2 (0.2%)	256 (25.6%)
total	1000	1000	1000

for the SVM models and greater than 87% for the decision tree models. The SVM models are marginally better than the decision-tree models.

To further qualify the discriminative power of the models, we performed a control experiment. We retrieved 1,000bp upstream sequences of 1,000 randomly chosen coding genes from each of the three species, and fragmented them using a sliding window of 250bp, with an increment of 50bp. Each segment was then tested by the discriminative models separately. The results from the SVM models are in Table 2. (The decision-tree models have comparable but slightly worse classification accuracies than the SVM models so the results are omitted.) Among 1,000 coding genes, 921 (92.1%) *C. elegans* genes, 630 (63.0%) *H. sapiens* genes and 693 (69.3%) *A. thaliana* genes have at least one segment predicted to be a pol-II promoter, and no segment classified as a pol-III promoter sequence.

Although some genes were predicted to contain segments of pol-III promoter sequences in their upstream sequences, (i.e., 16 (16%) *C. elegans* genes, 368 (36.8%) *H. sapiens* genes and 51 (5.1%) *A. thaliana* genes), only a handful of them were predicted to have more pol-III segments than pol-II segments (i.e., 8 (0.8%) *C. elegans* genes, 25 (2.5%) *H. sapiens* genes and 33 (3.3%) *A. thaliana* genes), which reflect the false prediction rates on these species. Based on these two sets of experiments, we can conclude that (1) the pol-II and pol-III promoters are separable from each other and are also distinguishable from random intergenic sequences, and (2) the quality of the discriminative models that we developed is sufficiently high.

2.2 MiRNA Genes Have Pol-II Promoters

To determine the promoter types of the known intergenic miRNA genes of the three model species, we conducted two experiments using the 3-class discriminative models that we developed. We considered separately the precursors (pre-miRNAs) and primary transcripts (pri-miRNAs) of the known miRNAs. We analyzed up to 2,000bp upstream sequences of these transcripts by following the same procedure used in the control experiments. That is, we segmented the upstream sequences using a sliding window of 250bp with a 50bp increment, and then tested each segment using the models.

We organized the experimental results in five categories. The first contained the upstream sequences in which at least one of the 250bp segments was classified as pol-II promoter and none of the rest was predicted as pol-III promoter. This class, called *definitive pol-II class*, provides the definitive evidence that miRNA genes are class-II genes. The second category had the sequences in which some of the segments were classified as pol-II and some as pol-III promoters, but pol-II segments was out numbered the pol-III segments. We call this category *possible pol-II class*, since we may simply classify a sequence to be pol-II promoter based on the majority prediction to its segments. The next category, called *possible pol-III class*, is similar to the second, but the number of pol-III segments was greater than the number of pol-II segments. The fourth category, called *definitive pol-III class*, had the sequences in which at least one segment was pol-III promoter but none of the rest was predicted as pol-II promoter. The last category, called *random class*, contained sequences with all segments classified as random promoters.

Table 3 shows the results on the known intergenic pre-miRNAs in the three species using the SVM models. The results from the decision-tree models were similar and are thus omitted here. We tested 70 *C.elegans*, 94 *H. sapiens* and 102 *A. thaliana* pre-miRNAs that are in the intergenic regions according to the genome annotation as of March, 2005. Among them, 64 (91.4%) *C. elegans*, 68 (72.3%) *H. sapiens* and 83 (81.4%) *A. thaliana* miRNAs have definitive pol-II class promoters. These species also have 1 (1.4%), 22 (23.4%) and 1 (1.0%) miRNAs that have possible pol-II promoters respectively. Combining the miRNAs in these two categories, we have 65 (92.4%) *C.elegans*, 90 (95.8%) *H. sapiens* and 84 (82.4%) *A. thaliana* miRNAs that have pol-II promoters. Importantly, none

Table 3. Classification results of miRNA genes using known pre-miRNAs and pri-miRNAs

promoter class	Pre-miRNAs			Pri-miRNAs	
	<i>C. elegans</i>	<i>H. sapiens</i>	<i>A. thaliana</i>	<i>H. sapiens</i>	<i>A. thaliana</i>
<i>Definitive pol-II</i> ^(a)	64 (91.4%)	68 (72.3%)	83 (81.4%)	9 (69.2%)	16 (84.2%)
<i>Possible pol-II</i> ^(b)	1 (1.4%)	22 (23.4%)	1 (1.0%)	1 (7.7%)	0
<i>Possible pol-III</i> ^(c)	0	1 (1.1%)	0	2 (15.4%)	0
<i>Definitive pol-III</i> ^(d)	0	0	0	0	0
<i>Random</i> ^(e)	5 (7.1%)	3 (3.2%)	18 (17.6%)	1 (7.7%)	3 (15.8%)
total	70	94	102	13	19

(a), (b), (c), (d) and (e) are the same as in Table 2

of the miRNAs was predicted to have a definitive pol-III promoter. Only one *H. sapiens* miRNA was predicted to have a possible pol-III promoter. Similar results, shown in Table 3, were obtained on *H. sapiens* and *A. thaliana* pri-miRNAs. However, the available pri-miRNAs are far fewer than pre-miRNAs. We were only able to find the pri-miRNA for *lin-4* in *C. elegans*, one of the first miRNAs reported, 13 pri-miRNAs for *H. sapiens* and 19 pri-miRNAs for *A. thaliana*. We expected the results based on pri-miRNAs to be more definitive than that from pre-miRNAs. However, it is difficult to draw a meaningful conclusion based on such limited samples. Nevertheless, as shown in Table 3, 9 out of 13 (69.2%) *H. sapiens* miRNAs and 16 out of 19 (84.2%) *A. thaliana* miRNAs were predicted to have definitive pol-II promoters, and importantly, none of them was classified to have definitive pol-III promoters.

In summary, the results of our experiments provided sufficient evidence that most miRNA genes, if not all, are class-II genes and have pol-II promoters.

2.3 Core Promoter Regions of MicroRNA Genes

It is difficult and time consuming to experimentally identify promoters. For *H. sapiens*, the promoters of two miRNA genes, *hsa-mir-23a~27a~24-2* [12] and *hsa-mir-371~372~373* [4], have been identified so far. The promoter of *hsa-mir-23a~27a~24-2* was identified by biological experiments [12] while the promoter of *hsa-mir-371~372~373* [4] was located by a comparative analysis of human and mouse. In this research, we developed a novel computational, sequence-centric method, named *common query voting* (CoVote), to identify the core promoter regions of miRNA genes. As described in the methods section, CoVote is based on the observation that the core promoters of miRNA genes, which are transcribed by pol II according to the results in this paper and [4,12,23], must share common sequence features, which can be exploited to identify the core promoter of a miRNA gene.

We evaluated the accuracy of our CoVote promoter identification method on the 40 *A. thaliana* miRNA genes experimentally studied in [23]. Our assessment criteria were fairly restrictive. We considered a prediction correct only if the predicted promoter included the transcription start site (TSS) of the tested

miRNA gene. Since multiple transcription start sites were reported for some of the known *A. thaliana* miRNA genes [23], we tested CoVote on 47 core promoters of 40 miRNA genes. As shown in Table 4, CoVote correctly identified 34 (72.3%) of the 47 known core promoter sequences. For 32 out of 40 (80%) *A. thaliana* miRNA genes, CoVote predicted at least one core promoter region correctly. As a comparison, TSSP, one of the best promoter prediction methods specific for plants [17], correctly identified 31 (66%) promoters from 29 (72.5%) miRNA genes on the same set of sequences. This analysis showed that our new promoter prediction method is fairly accurate.

Using CoVote, we predicted putative core promoters of most known miRNA genes of the three species that we studied, as shown in Supplementary Materials. Specifically, we predicted 69 (98.6%) promoters for 70 *C. elegans* miRNA genes, 91 (96.8%) promoters for 94 *H. sapiens* miRNA genes, and 49 (96.1%) for the 51 *A. thaliana* miRNA genes whose promoters have not been identified by experimental approach. The detailed loci information of these promoters is also available in the Supplementary Materials.

Table 4. Promoter prediction results of known promoter sequences of class-II genes. (a): # of promoters correctly predicted, (b): # of miRNA genes at least one of whose core promoter regions was correctly predicted.

	# promoters correct ^(a)	total	# genes correct ^(b)	total
CoVote	34 (72.3%)	47	32 (80%)	40
TSSP	31 (66.0%)	47	29 (72.5%)	40

3 Discussion

3.1 MicroRNA Genes Have Class-II Type Promoters

The results on three widely separated species that we examined provided clear evidence that intergenic miRNA genes have class-II type promoters, suggesting that miRNA genes are transcribed by pol II. None of 2,000bp upstream sequences of pre-miRNAs was predicted to belong to definitive pol-III promoter class, whereas at least 72% of these sequences were predicted to be in the class of definitive pol-II promoters (Table 3). Therefore, our computational results are consistent with and complement the conclusion that Lee *et al*, Houbaviy *et al* and Xie *et al* drew based on experimental means [4,12,23].

Some pre-miRNAs, particularly those in *H. sapiens*, were classified to be in possible pol-II class. One pre-miRNA and two pri-miRNAs of *H. sapiens* were even predicted to be in the possible pol-III class in Table 3. This may be because the discriminative models that we developed, like many other computational models, are not perfect and the current *H. sapiens* genome annotations are incomplete. Furthermore, even though most miRNA genes are class-II genes, there may be possible exceptional cases where some specific miRNA genes are transcribed by pol III. Such exceptional cases exist in other non-coding genes. For instance, most snRNA genes are transcribed by pol II, while U6 snRNA genes rely on pol III.

3.2 Core Promoter Regions of miRNA Genes

The existing promoter prediction methods were developed for coding genes and used various knowledge specific to known coding gene promoters, such as those with and without the TATA-box. These methods do not seem to be directly applicable to miRNA genes, because little is known about the genomic structure of these relatively new genes. Although the promoters of miRNA genes have some similar or even the same features as the known promoters of class-II genes, they may have their own unique features that have not been observed in the known promoters of the class-II genes.

Our new promoter identification method, CoVote, included both the motifs extracted from the core promoters of pol-II coding genes and similar features among the promoters of most miRNA genes. Our new method is effective; we were able to identify putative core promoters for most known miRNA genes of the three species that we studied. Control experiments on the known promoters of 40 *A. thaliana* miRNA genes showed that the accuracy of our promoter identification method is better than the best promoter prediction method for plant. Importantly, our method can be flexibly applied to multiple species, while most existing promoter prediction methods are restricted to single species.

We note that the promoters of two miRNA genes, *hsa-mir-23a~27a~24-2* [12] and *hsa-mir-371~372~373* [4] identified in previous studies were all included in the corresponding promoter regions predicted in this research as shown in Supplemental Materials. On the other hand, more studies are required to fully understand the features and structures of core miRNA promoters. The putative promoters from our study can serve the need of future biological verification and analysis. For example, assays based on recombination with reporter gene can be used to verify the essential promoter regions. To facilitate such bioassay experiments or other analysis in the future, we listed in Supplemental Materials the location information of the putative core promoters of known miRNA genes in the three model species.

3.3 Future Improvements

Our main results in this paper depend on the accuracy of the discriminative models of pol-II and pol-III promoters that we developed. Although we have shown by cross-validation and control experiments that our decision-tree and SVM based discriminative models are sufficiently accurate, their quality can be further improved in several ways when additional data are available.

First, in our current discriminative models, we combined the promoters of U6 snRNA, 7SL and 7SK RNA genes with the promoters of tRNA genes to form a training set for pol-III promoters. Although U6 snRNA genes are transcribed by pol III, the promoters of these genes and protein coding genes share many similarities, such as TATA box [19,21]. Thus, ideal methods should include a discriminative model to finely classify the U6 snRNA promoters and protein coding gene promoters. However, the known U6 snRNAs with available core promoters are very few, and we only collected 8 in *C. elegans*, 4 in *H. sapiens*

and 7 in *A. thaliana* (see Supplementary materials). As a result, a model based on such few training data will not be reliable.

Second, the discriminative models of different type promoters for *A. thaliana* can be improved. The upstream regions of 256 (25.6%) out of 1,000 coding genes and 18 (17.6%) among 102 pre-miRNAs were predicted to be random sequences. A possible reason is that there were relatively fewer known pol-II core promoters for *A. thaliana*. In our study, we included core promoter sequences from 44 dicotyledonous and 7 monocotyledonous plants (see Supplementary materials). Therefore, the training set did not precisely characterize *A. thaliana* pol-II promoters. In contrast, the models for *C. elegans* and *H. sapiens* were more accurate where training data were exclusively from these species. We expect the quality of the models for *A. thaliana* to improve using more verified pol-II core promoters.

Finally, our method for predicting miRNA promoters can be improved. Our method failed to identify the promoters of a small number of miRNA genes. A possible reason is that the promoters of these genes are not similar to either the promoters of many known promoters or the promoters of most other miRNA genes. Including more known promoters in the training sets and developing a more sensitive model would be two possible ways to improve our method.

In summary, we have studied the promoters of the known intergenic miRNA genes in three model species, *C. elegans*, *H. sapiens* and *A. thaliana*. The genome-wide evidence from these three species showed that most miRNA genes, if not all, have the same type of promoters as protein-coding genes. Moreover, with a new promoter identification method, we also located the core promoter regions of most known miRNA genes. We expect our results on the putative promoters to be useful for future miRNA prediction and for elucidating transcriptional regulation of miRNA genes.

4 Material and Methods

4.1 Data Sets

To train discriminative models of RNA polymerases, we used experimentally verified core pol-II promoters of *C. elegans*, *H. sapiens*, and *A. thaliana*. The numbers of these promoter sequences are listed in Supplemental Table 1. The data were downloaded from the web as of March, 2005. The *C. elegans* core pol-II promoters were retrieved from *C. elegans* promoter database (CEPDB) (<http://rulai.cshl.edu>). The *H. sapiens* core pol-II promoters were downloaded from Eukaryotic Promoter Database (EPD), Release 83 (<http://www.epd.isb-sib.ch>). The plant core pol-II promoters were obtained from Plant Promoter Database (PlantProm) (<http://mendel.cs.rhul.ac.uk>). The pol-III promoter set included the promoter sequences of tRNAs, U6 snRNAs, 7SL RNAs and 7SK RNAs. The promoter of each tRNA covered the complete coding region of the tRNA and its upstream sequence with a total length of 250bp. The promoters of U6 snRNA, 7SL RNA and 7SK RNA included 200bp upstream sequences and 50bp downstream sequences, relative to their TSSs. The sequence of these ncRNAs were downloaded from ncRNA database,

<http://noncode.bioinfo.org.cn>. We generated 1,000 random sequences of 250bp long to represent intergenic sequences other than pol-II and pol-III core promoter sequences. For each species, we used nucleotide composition of intergenic regions of the genome to generate the sequences. We did not use intergenic sequences from a genome for this purpose because it is difficult to ensure the intergenic sequences not to overlap with real promoter regions.

Only the intergenic miRNA genes were considered in our study. We retrieved the upstream sequences of pre-miRNAs of *C. elegans*, *H. sapiens* and *A. thaliana* according to their annotations. The upstream sequence of a pre-miRNA was obtained as follows: First, when a pre-miRNA and its upstream gene were unidirectional (same direction), if the distance between them was longer than 2,400bp, the 2,000bp sequence upstream the pre-miRNA was retrieved; otherwise (the distance was shorter than 2,400bp), the sequence between 400bp downstream of the upstream gene and the precursor was used. Second, when a pre-miRNA and its upstream gene were convergent (opposite directions), if the distance between them was longer than 4,000bp, the 2,000bp sequence upstream of the precursor was obtained; otherwise, the sequence from the precursor and the middle point between the upstream gene and the precursor was retrieved. The 250bp segments of each upstream sequence were prepared using a 250bp sliding window with a 50bp increment.

For each species studied, 1,000 randomly chosen coding genes were used in the control experiment, and 1,000-bp upstream sequences of these genes were prepared in the same way as for pre-miRNAs. These sequences were obtained from RSA Tools (<http://rsat.ulb.ac.be/rsat/>).

4.2 Subsequence Feature Extraction

Our overall approach depends on building accurate discriminative models of promoter classes, which in turn rely on sequence features. We may simply use all possible k -mers, with reasonable values of k , from the promoter regions as such features. However, not all k -mers have the same amount of information, and the number of k -mers increases exponentially with k . The key is then to find a sufficient number of statistically over-represented motifs in the sequences of interest.

We used our WordSpy algorithm [20] for finding significant motifs for several reasons. First, statistical modeling and word counting methods are integrated in WordSpy so that it is able to build a dictionary of a large number of statistically significant motifs. WordSpy adopts a strategy of steganalysis, which is a technique for discovering hidden information and patterns from sequences, so that it does not have to rely on additional background sequences and is still able to find motifs of nearly exact lengths. In the context of English, for example, substrings ‘the’, ‘ther’ and ‘there’ are usually over-represented with respect to random strings. However, WordSpy is able to determine that ‘the’ and ‘there’ are significant while ‘ther’ is not; using the steganalysis strategy, the algorithm is able to detect that the significance of ‘ther’ is merely due to ‘there’. The detail of the algorithm and its results on a large English stegoscript

and the motifs in the promoters of budding yeast genes are available in [20] and <http://cic.cs.wustl.edu/wordspy/>.

4.3 Decision Trees and Support Vector Machines for Discriminative Models

With sequence motifs extracted, we formulated a promoter sequence by a vector of these features, where an entry in the vector was the number of occurrences of a motif in the sequence. We then applied supervised machine learning methods to build discriminative models to distinguish different classes of sequences based on the features. Specifically, we used the well studied and applied methods of decision trees [14] and support vector machines (SVMs) [16]. Both methods have high classification accuracies, and were used these methods to model the pol-II and pol-III promoters.

The SVM implementation that we used was from WEKA software package [5]. We tested linear, polynomial and radial kernels with default parameters [16]. The cross-validation accuracies of the polynomial and radial kernels were slightly better than that of the linear kernel. We used the linear kernel due to its simplicity. We used the C4.5 algorithm implementation [14] in WEKA for decision trees with its default settings. To prevent over-fitting, we required each leaf node to have at least 5 instances (sequences).

Modeling pol-II and pol-III promoters is a three-way classification problem for separating pol-II core promoters, pol III core promoters and random sequences. The decision-tree method can directly solve this three-way classification problem, while the standard SVM method supports only pairwise classification. We used a majority voting mechanism to extend SVMs to multi-class classification. For a problem of n classes, we built a classifier for each of the $n(n-1)/2$ class pairs, and selected the final classification by the majority vote of all classifiers.

The accuracy of a discriminative model was estimated using a ten-fold cross validation. In this process, the training data were uniformly partitioned into ten roughly equal-size subsets. Each subset was then used in turn as a test set to estimate the prediction accuracy of the model that was built with the other nine subsets. The average accuracy of these tests was then used as the final measure. To measure prediction quality, we calculated sensitivity, specificity and overall accuracy for each type of sequences. The sensitivity for pol-II promoters (III, respectively) is defined as the ratio of the number of correctly predicted pol-II (III, respectively) sequences versus the total number of pol-II (III, respectively) sequences tested. The specificity is defined as the ratio of the number of correctly predicted pol-II (III, respectively) sequences versus the total number of predicted pol-II (III, respectively) sequences. The overall accuracy is defined as the number of correctly predicted sequences versus the total number of sequences tested.

4.4 Finding Core miRNA Promoters

Our approach, which we called *common query voting*, shorthanded as CoVote, is based on the following understanding on the miRNA gene promoters. MiRNA

genes have the same type of promoters as other class-II genes, as shown in this paper and in [4,12,23]. Additionally, sequence features in the core promoters of miRNA genes are over-represented, with respect to random sequences that are generated with the same nucleotide compositions of intergenic sequences. Moreover, compared with other upstream regions, core promoters would be the most similar upstream regions among most, if not all, miRNA genes. Although the promoters of miRNA genes have some similar or even the same features as the identified promoters of the known class-II genes, they may have their own unique features that have not been discovered. Compared with the existing promoter prediction methods, CoVote includes not only the sequence features extracted from the core promoters of protein coding genes, but also possibly similar features among the promoters of miRNA genes themselves. In other words, CoVote takes into account the features that the training instances have as well as potential common features in many query instances. The CoVote algorithm runs as follows:

- **Model training step:** Train a two-class decision tree model with known pol-II promoters as positive examples and randomly generated sequences as negative training examples, in a way similar to the previous three-class models.
- **Classification step:** Apply the two-class model to the upstream sequences of miRNA genes, fragmented into overlapping 250bp segments as described previously. Each segment is predicted to be either pol-II promoter or random sequence by the tree at one of its leaf nodes. The classification of a segment corresponds to following a path from the root to a leaf node in the tree, and the nodes on the path represent the sequence motifs used. Therefore, the decision tree model provides a mechanism for identifying the segments that must belong to the same core promoter class using the same set of sequence motifs.
- **Scoring step:** Each leaf node is assigned a weight that is equal to the number of miRNA genes that have at least one upstream segment classified to be pol-II promoter at that leaf node. Then, the score of each upstream segment that has been predicted to be pol-II promoter is the weight of the leaf node at which it is classified. This weighting scheme explicitly take into account the similarities among the putative promoters of miRNA genes themselves. The weight of a leaf node reflects how many upstream sequences follow the rule specified by the path from the root node to this leaf node. Since the weight of a segment can thus be viewed as a vote of other similar segments, hence we name our method *common query voting* or CoVote for short.
- **Putative promoter identification step:** For a miRNA gene, consecutive segments of non-zero score in its upstream sequence are combined. The score of the combined subsequence is the sum of the scores of these consecutive segments. These subsequences are then taken to be the putative core promoter regions of the miRNA gene according to a cutoff score that is set by the user. Note that a particular miRNA gene may be predicted to have multiple putative promoter regions.

Supplementary Materials: The supplementary materials, which are available at <http://cic.cs.wustl.edu/microrna/promoters.html>, contain additional supporting results and data.

Acknowledgement. This research was funded in part by NSF grants EIA-0113618 and IIS-0535257.

References

1. D.P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, Jan 2004.
2. M.T. Bohnsack, K. Czaplinski, and D. Gorlich. Exportin 5 is a RanGTP-dependent dsRNA-binding protein that mediates nuclear export of pre-miRNAs. *RNA*, 10(2):185–91, Feb 2004.
3. J.C. Carrington and V. Ambros. Role of microRNAs in plant and animal development. *Science*, 301(5631):336–8, Jul 2003.
4. H.B. Houbaviy, L. Dennis, R. Jaenisch, and P.A. Sharp. Characterization of a highly variable eutherian microRNA gene. *RNA*, 11:1245–57, 2005.
5. Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publisher Inc, 1999.
6. M. Lagos-Quintana, R. Rauhut, W. Lendeckel, and T. Tuschl. Identification of novel genes coding for small expressed RNAs. *Science*, 294(5543):853–8, Oct 2001.
7. N.C. Lau, L.P. Lim, E.G. Weinstein, and D.P. Bartel. An abundant class of tiny RNAs with probable regulatory roles in *Caenorhabditis elegans*. *Science*, 294(5543):858–62, Oct 2001.
8. R.C. Lee and V. Ambros. An extensive class of small rnas in *caenorhabditis elegans*. *Science*, 294(5543):862–4, Oct 2001.
9. R.C. Lee, R.L. Feinbaum, and V. Ambros. The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell*, 75(5):843–54, Dec 1993.
10. Y. Lee, C. Ahn, J. Han, H. Choi, J. Kim, J. Yim, J. Lee, P. Provost, O. Radmark, S. Kim, and V.N. Kim. The nuclear RNase III Drosha initiates microRNA processing. *Nature*, 425(6956):415–9, Sep 2003.
11. Y. Lee, K. Jeon, J.T. Lee, S. Kim, and V.N. Kim. MicroRNA maturation: stepwise processing and subcellular localization. *EMBO J*, 21(17):4663–70, Sep 2002.
12. Y. Lee, M. Kim, J. Han, K.H. Yeom, S. Lee, S.H. Baek, and V.N. Kim. MicroRNA genes are transcribed by RNA polymerase II. *EMBO J*, 23(20):4051–60, Oct 2004.
13. E. Lund, S. Guttinger, A. Calado, J.E. Dahlberg, and U. Kutay. Nuclear export of microRNA precursors. *Science*, 303(5654):95–8, Jan 2004.
14. J.R. Quinlan. *C4.5: Programs for Machine Learning*. MK, 1993.
15. B.J. Reinhart, F.J. Slack, M. Basson, A.E. Pasquinelli, J.C. Bettinger, A.E. Rougvie, H.R. Horvitz, and G. Ruvkun. The 21-nucleotide *let-7* RNA regulates developmental timing in *Caenorhabditis elegans*. *Nature*, 403(6772):901–6, Feb 2000.
16. B. Scholkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.
17. I.A. Shahmuradov, A.J. Gammerman, J.M. Hancock, P.M. Bramley, and V.V. Solovyev. PlantProm: a database of plant promoter sequences. *Nucleic Acids Res*, 31(1):114–7, Jan 2003.

18. S.T. Smale and J.T. Kadonaga. The RNA polymerase II core promoter. *Annu Rev Biochem*, 72:449–79, 2003.
19. F. Waibel and W. Filipowicz. U6 snRNA genes of Arabidopsis are transcribed by RNA polymerase III but contain the same two upstream promoter elements as RNA polymerase II-transcribed U-snRNA genes. *Nucleic Acids Res*, 18(12):3451–8, Jun 1990.
20. G. Wang, T. Yu, and W. Zhang. WordSpy: identifying transcription factor binding motifs by building a dictionary and learning a grammar. *Nucleic Acids Res*, 33(Web Server issue):W412–6, Jul 2005.
21. Y. Wang and W.E. Stumph. RNA polymerase II/III transcription specificity determined by TATA box orientation. *Proc Natl Acad Sci U S A*, 92(19):8606–10, Sep 1995.
22. B. Wightman, I. Ha, and G. Ruvkun. Posttranscriptional regulation of the heterochronic gene *lin-14* by *lin-4* mediates temporal pattern formation in *C. elegans*. *Cell*, 75(5):855–62, Dec 1993.
23. Z. Xie, E. Allen, N. Fahlgren, A. Calamar, S.A. Givan, and J.C. Carrington. Expression of Arabidopsis miRNA genes. *Plant Physiol*, 138(4):2145–54, Aug 2005.

A Comprehensive Kinetic Model of the Exocytotic Process: Evaluation of the Reaction Mechanism

Aviv Mezer¹, Eran Basis¹, Uri Ashery², Esther Nachliel¹, and Menachem Gutman¹

¹Laser Laboratory for Fast Reactions in Biology, Department of Biochemistry,
Tel Aviv University

²Department of Neurobiochemistry, Tel Aviv University

Abstract. This study presents a comprehensive, quantitative description of the exocytotic process that has both analytic and predictive powers. The model utilizes strict chemical formalism and is based on a set of equilibria between the various SNARE proteins, their complexes and the reaction which free Ca^{2+} ions. All these reactions are linked by first and second order rate constants. With the proper set of rate constants, which were selected by a systematic search in a multi dimensional parameter space, the model reconstructs the fusion dynamics as recorded under divergent experimental protocols: the effect of repeated depolarization, recovery after depletion of the vesicular pools of the cell, and over-expression or knockout of specific proteins. The model provides a detailed scenario of the maturation process, where the vesicles progress from the early steps of the SNARE complex formation up to the last event of the vesicles' fusion with the cells' plasma membrane. The dynamics of each intermediate enable us to describe the experimental result in terms of overall flow, where upstream intermediates are mobilized during the progression of the reaction.

1 Introduction

Exocytosis is a well coordinated process, by which intracellular trafficking vesicles fuse with the plasma membrane. In neurons and in neuroendocrine cells, exocytosis is a multi-step process that is regulated by calcium and can be described as a sequential process of vesicle maturation between defined states which are termed as 'pools'. At the molecular level, exocytosis is mediated by a sequence of interactions between cytosolic, vesicular and plasma membrane proteins. The process can be measured with chromaffin cells and, in a few large synaptic terminals, at a millisecond accuracy using membrane capacitance measurements [1]. These studies have revealed several kinetic components in the process of exocytosis and have elucidated the involvement of several synaptic proteins [2-9] and the sequence of the reactions between them, even though some steps are still not well defined. Recently, we presented a kinetic model [10], based on a strict chemistry kinetics formalism, that accurately reconstructs the exocytosis as measured under various experimental protocols, where the chromaffin cells were challenged by various stimuli before exocytosis was induced. The model linked the known interactions between the SNARE proteins into an array of chemical equilibria that were converted into a set of differential rate

equations. The integration of these equations, with the proper values assigned to the rate constants of each partial reaction, generated the dynamics of the system (for details see [10]).

The present study tests the ability of the model to reconstruct some fine features of the dynamics, which had not been previously treated. This refinement of the model expands its scope, thus rendering it with predictive power. We suggest that the present system can be used as a standard representation of the exocytotic process, which may evolve into a diagnostic tool revealing the sites where the sequence of events is fundamentally impaired.

2 Methods

The partial reactions of the fusion model: The reaction mechanism associated with the fusion process is summarized in Scheme I, and was fully elaborated in [10]. In brief, the reaction flow chart represents the mechanism as a series of consecutive reactions linked to each other by the strict rules of chemical kinetics; the velocities are the product of the reactant concentrations times of the rate constants. In this formalism, the equilibrium constant is equal to the ratio of the forward and backward rate constants. The conversion of these reactions into a set of coupled ordinary differential equations generates a system that upon integration can reproduce experimental results. In order to reconstruct the observations, both the concentrations and the rate constants are used as adjustable parameters; however, in the case where the value was published, the search was limited to a narrow range centered around the published value. All other adjustable parameters were searched over a wide range, where the upper limit is the value estimated by the Debye-Smoluchowski equation [11]. The concentrations are expressed in molar units, both for the soluble reactants (munc13, Ca^{2+}) and for the vesicles. To account for the fact that each vesicle can have more than a single copy of a given protein, the multiplicity of sites was embodied into the rate constant of the reaction according to the equations of Berg and Purcell [12] (For details see [10]). The application of the Genetic Algorithm as a search strategy yielded discrete values for all the adjustable parameters needed to reproduce a large set of observation, which were gathered under varying initial conditions [13].

3 Results and Discussion

The exocytosis is a complex cellular reaction and its reconstruction must comply with the 'history' of the cell, i.e. treatments that precede the actual fusion step. Accordingly, the reconstructions of the fusion events were allowed to follow three distinct reaction phases which correspond with the experimental system:

The resting state of the cells

Before the initiation of the measurements, the intra-cellular [Ca^{2+}] is very low (~50 nM). During this period, all SNARE proteins can interact with each other and attain equilibrium with the limited concentration of Ca^{2+} ions. On modeling the dynamics, we introduced a 10 min. period in which the equations were propagated in time at a

constant low Ca^{2+} concentration, 50 nM. Extension of the resting phase beyond the 10 min. time range hardly affected the results.

The pre-pulse phase

Before measuring the release dynamics, it is common to raise the intra-cellular Ca^{2+} concentration to ~ 300 nM for a period of ~ 2 min. This step has a pronounced effect on the shape and magnitude of the release dynamics. For the modeling of the process, the final concentrations of reactants of the resting phase were taken as the initial values and the system was propagated over time for 120 seconds.

The fusion of the vesicles

For the last phase of the exocytosis, the intracellular Ca^{2+} concentration is raised by photo-chemical reactions to a level of $20\text{--}30$ μM . In the simulation of the fusion step, the calculations were initiated using the final values derived in the pre-pulse phase, and the free Ca^{2+} concentration was set at 30 μM . Under these conditions, the system was propagated for 5 seconds. The fusion was expressed as the number of vesicles that fused with the membrane.

3.1 Reconstruction of the Fusion Dynamics

The experimental signal, measured with the WT cells, together with the reconstructed dynamic, were calculated as described in [10] and presented in Fig.1. The quality of the reconstruction of the experimental signal is self-evident, by the overlaying of curves a (experimental) and b (theoretical). The fusion in the WT cells proceeds by two parallel pathways (see Scheme I): Path I where synaptotagmin-I (SytI) is the Ca^{2+} sensor and Path II where the alternate Ca^{2+} sensor (Syt*) is the Ca^{2+} sensor. The numeric reconstruction allows us to present the process through its constituents (Fig.1, curves C and d), where the fusion dynamics are operating through the two

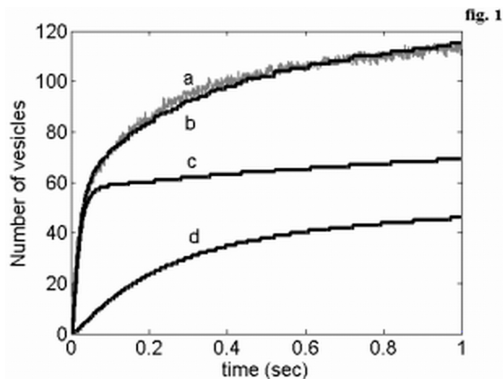
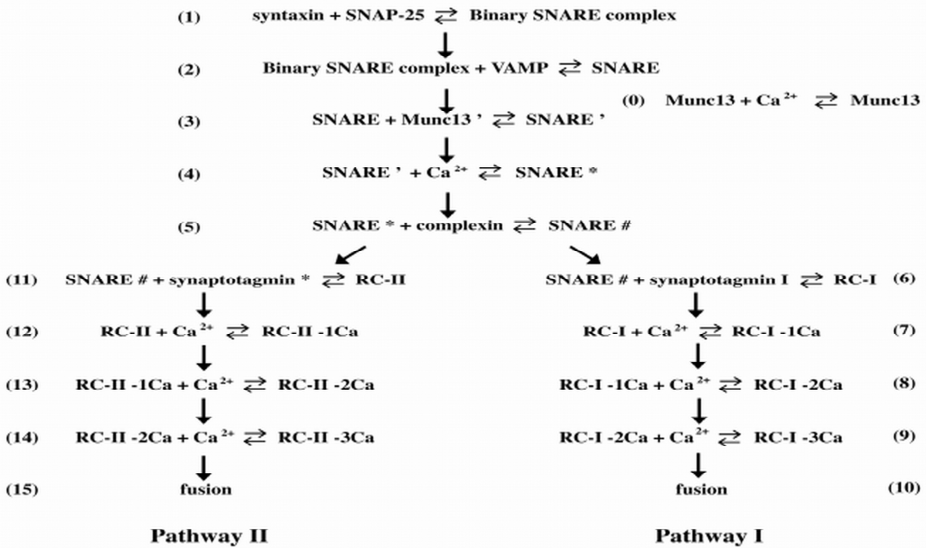


Fig. 1. Reconstruction of the standard experimental signal. Curves a and b depict the measured and reconstructed signal of the WT cells, respectively. Curves c and d depict the relative contributions of Path I and Path II to the reconstructed curve. The sum of the two parallel reactions (curve b) can reconstruct the experimental signal (a).



Scheme I. The sequence of reactions needed to reconstruct the fusion dynamics of the Ca^{2+} triggered exocytotic process. A sequential number, the nature of the reaction, the equilibrium and rate constants characterize each reaction. For the sake of brevity, the intermediate complexes are defined in the scheme and shall be referred to by the acronyms (munc13'; SNARE'; SNARE*; SNARE[#]; RC-I and RC-II.) as defined in the Scheme. For details see [10].

Ca^{2+} sensors. Path I contribute the fast release phase that dominates in the early phase of the fusion (~50-100 ms), while Path II, operating on a different time scale, contributes mostly to the slow element of the burst. The two pathways share common upstream reactions, and thus compete with each other. Accordingly, the understanding of the mechanism cannot be limited to the reconstruction of the final fusion reactions. Therefore, we investigated all the reactions, starting with the resting phase events and ending with the milliseconds dynamics of the fusion.

3.2 The Dynamics of the Unobserved Steps of the Phases of the Reaction

The detailed dynamics of the intermediates during the three time frames are presented in Fig.2, (A, B and C). The dynamics are presented in small frames, each of which corresponds with a certain intermediate defined in the reaction scheme. The ordinate is expressed in the number of vesicles and the abscissa is in time. For clarity, the frames are arranged in the same order as the intermediates in the reaction scheme.

Fig.2 A depicts the dynamics during a 10 min. period, where the vesicles are maturing in the presence of basal Ca^{2+} concentration, 50 nM (resting phase). During this state, three reaction steps had attained a state of equilibrium: one is the activation of munc13 that accumulates as munc13' to about 10% of its total content. The second one is the formation of the binary SNARE complex. This reaction is regulated by upstream steps, which kinetically had not yet been defined (binary SNARE complex formation). The binary SNARE complex reacts with VAMP to form the ternary

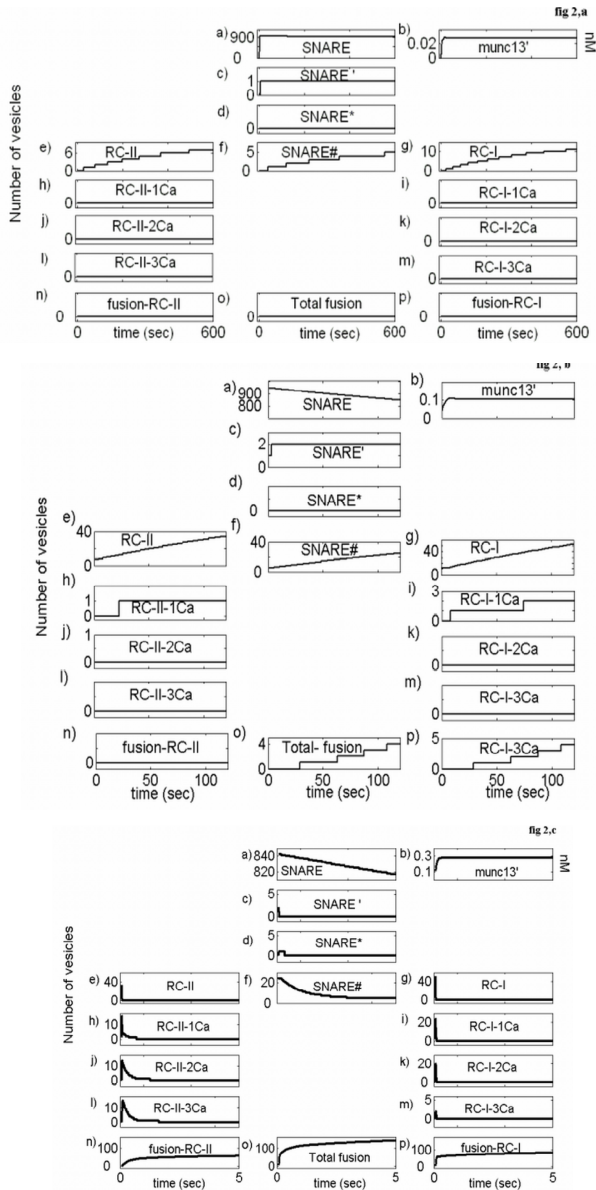


Fig. 2. Detailed simulation of the fusion process that demonstrates the evolution of the various intermediates from the resting phase (frame A), during the pre-pulse (frame B) up to the fusion reactions that are initiated by the high Ca^{2+} pulse (frame C). Each rectangle depicts the variation of each intermediate, expressed in number of vesicles at the state of maturation as marked in the boxes. Munc13 is given in nM. Please note that the last frame, corresponding with the fusion step, is the summation of the two pathways.

SNARE, which accumulates to its maximal content in a few seconds. During the resting phase, there is hardly any further maturation of the vesicles beyond the ternary-SNARE state. Inspection of the dynamics presented in frame A reveals that, out of the ~1000 docked vesicles, only ~20 had progressed in their maturation. About 5 vesicles are in the SNARE[#] state, ~10 are in the RC-I and ~7 in RC-II. Extension of the resting dynamics for a longer time (20 min) did not affect the kinetic features of the dynamics.

The dynamics of the intermediates during the pre-pulse phase, where the Ca²⁺ level was raised to ~300 nM, are depicted in Fig.2 B. During this period, munc13 is further activated and a steady level of 37% is established within less than 15 seconds. Thus, the rate of SNARE maturation is controlled by the level of active munc13 (munc13'). The higher level of munc13' initiates a further flow of the vesicles along the maturation axis, and some ~100 vesicles proceeded in their maturation, accumulating in three reservoirs: the SNARE-complexin (SNARE[#] ~30 vesicles) and the two complexes with the Ca²⁺ sensors: SytI (RC-I; ~50 vesicles) and Syt* (RC-II; ~30 vesicles). The concentrations of the two accumulated complexes are not identical, representing a higher stability of RC-I. The prevailing Ca²⁺ concentration in the pre-pulse phase (~300 nM) is too low to drive a massive fusion reaction. Yet, there is a minor leak initiated by low probability encounters between Ca²⁺ ions and the fusion compatible vesicles (RC-I), ending by fusion events (see the right bottom rectangle in frame B). At higher pre-pulse Ca²⁺ concentrations, the leak is enhanced, in accordance with the experimental results. The modeling of the systems replicates the leak, assigning it to a preferential reactivity of the Path I intermediates.

Fig.2 C depicts the events during the fusion phase. The response of the system during the Ca²⁺ pulse can be classified into two time-domains: a fast and a delayed one. The fast events dominate during the first ~200ms. Within this time frame, the vesicles that had accumulated during the pre-pulse phase in the two reservoirs (RC-I and RC-II) are rapidly evolving into downstream intermediates. However, the fast depletion of the reservoirs does not coincide with the release dynamics; the products of the RC-I-Ca²⁺ intermediates rapidly evolve and fuse with the membrane, with minimal accumulation of Ca²⁺-containing intermediates. On the other hand, the vesicles in Path II exhibit different dynamics; the initial steps are as fast as in Path I. But, as the rate-limiting step of Path II is the fusion reaction, which is ~100 times slower than that of path I, the intermediates between RC-II and the fusion complex accumulate and the release from path II extends in time up to ~1 second. The delayed response of the cell to the high Ca²⁺ pulse is characterized by another reaction, termed the 'sustained' release. The activation of the residual munc13, which takes only a fraction of a second, accelerated the mobilization of the remaining SNARE complex, thus enhancing their maturation into downstream intermediates. The consumption of the SNARE intermediate appears to be linear and its rate is comparable to the "sustained" phase of the fusion, the one that appears as a linear slow release. This suggests that the maturation of the SNARE is the rate-limiting step during the later phase of the exocytosis process.

Examination of the dynamics of fusion indicated that, with time, the rate-limiting step of the fusion mechanism drifts upstream. In the initial phase, the rate-limiting step is equated with the reaction of the RC-I and RC-II with free Ca²⁺ ions, a reaction that empties the two reservoirs. Later, some two seconds after initiation, the

rate-limiting step had drifted upstream and the reaction between SNARE[#] and SytI and Syt* limits the availability of RC-I and RC-II to react with the prevailing Ca²⁺ ions. Finally, during the linear sustained fusion, the rate-limiting step is an upstream event, identified as the maturation of the SNARE by the action of Munc13'.

3.3 The Lag Phase of the Fusion

Kinetic measurements of the most early event following the high Ca²⁺ pulse reveal a distinct delay (lag phase) in the fusion reaction. The length of the lag phase is shortened at an increasing Ca²⁺ concentration [14, 15]. The same phenomenon is reconstructed by the model, with no need to modulate any of the adjustable parameters, and is presented in Fig. 2.

The left frame depicts the dynamics of the intermediates of path I during the first 50 ms. Following the high Ca²⁺ pulse, the RC-I state of the vesicles is rapidly depleted, with subsequent accumulation of the first Ca²⁺ complex RC-I-1Ca. This complex is converted into the higher Ca²⁺ complexes RC-I-2Ca, RC-I-3Ca and the fusion products. The four sequential steps of the reaction pathway consist of three pseudo-first order rate constants, thus explaining why the overall dynamics and the lag phase are affected by [Ca²⁺]. For a given Ca²⁺ concentration of ~ 30 μM, as used by Voets [15], the apparent rate constants ($k_{on} * [Ca^{2+}]$) characterizing the reaction in path I are 1700, 420 and 300 s⁻¹, which are all smaller than the final fusion step which has a true first order rate constant of 2700 s⁻¹. Thus, the last step is faster than all others and the reaction is pulled forward. However, as the last complex RC-I-3Ca fuses at a faster rate than its formation, this complex hardly accumulates and the lag time appears to stretch over the time frame where RC-I-1Ca and RC-I-2Ca intermediates are formed. The right frame reconstructs the dynamics of the final steps of fusion through path II. The shape of the curves is similar to those in frame A, except that, due to the rate limiting step at the end of the path, the accumulation of intermediates is more pronounced and the dynamics are stretched over a longer period.

3.4 The Versatility of the Model

The present state of the model is compatible with the current data regarding the interactions between 7 synaptic proteins and calcium. Yet, in the fast-evolving field of exocytosis, it is very likely that the presence and role of more proteins will be elucidated. It is therefore essential to investigate whether the model can be refined and expanded when new information is revealed. Most of the rate constants described in this model are slower than the estimated diffusion-controlled values and thus favor this possibility. In these cases, we assume that the actual reaction step consists of more than a single one, and the values determined by the analysis are of the rate-limiting step. The hidden steps can either be a conformational change of the protein, a reaction of the indicated complex with other (still unidentified) protein(s), or even oligomerization of the complex/protein. A test as to whether the model can be expanded to accommodate major reaction steps can definitely be an investigation of the inverse case: Will the model sustain the omission of one step by incorporating the dynamics of the missing step into other rate constants? For this reason, we deliberately mutilated the reaction scheme by removing an established step: the reaction between SNARE* with complexin, with the assumption that SNARE[#] is a

direct product of reaction 4, the activation of SNARE' by the Ca^{2+} ions. A search for a solution that would reconstruct the observation was carried out, using as a model the reaction sequence lacking the complexin involvement. The search yielded a new solution, with practically the same values for all the reaction steps, except for the one where a reacting component had been removed. In the new system, the omission of complexin from the reaction sequence was ameliorated by a 50-fold increase of the Ca^{2+} affinity of reaction 4, reducing K_{eq} from 560 nM to 10 nM. This search, which was a test of whether the system is flexible enough for the missing step to be ameliorated through local modulation of the rate constants, demonstrates that the model is sufficiently robust to suffer a shortening of the reaction sequence, and still reconstructs the experimental observation. In the same sense, we can project that ,in the future, upon addition of new reactions to the model, according to new experimental results, the model integrity would not be affected.

To sum up, we wish to suggest that the present model, of which each step consists of a well-defined chemical reaction, can be elaborated into a diagnostic tool, revealing the sites where the sequence of events is fundamentally impaired. There are quite a few neurodegenerative diseases, where the exocytotic apparatus seems to be impaired. The model may be used for the definition of the reaction steps, where the properties of the reaction had markedly changed, suggesting that the malfunction may be associated with the very same step.

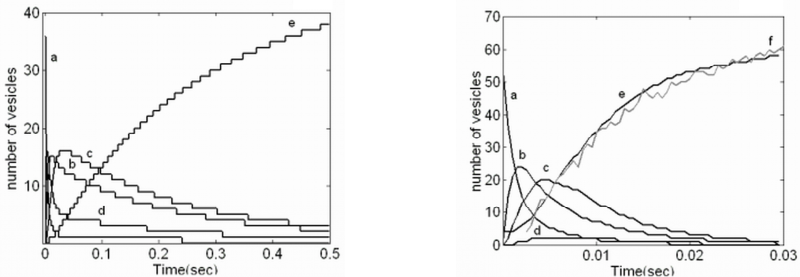


Fig. 3. A and B. Detailed presentation of the last steps of the fusion mechanism: the reaction of the Ca^{2+} sensors with the Ca^{2+} ions. Left: Superposition of the dynamics of the intermediates along path I: RC-I, RC-I-1Ca, RC-I-2Ca, RC-I-3Ca, its fusion product. Right: Superposition of the dynamics of the intermediates along path II: RC-II, RC-II-1Ca, RC-II-2Ca, RC-II-3Ca, its fusion product. All states are given as the number of vesicles.

Acknowledgments. The heading should be treated as a 3rd level heading and should not be assigned a number.

References

1. Xu, T., et al., Multiple kinetic components of exocytosis distinguished by neurotoxin sensitivity. *Nat Neurosci*, 1998. **1**(3): p. 192-200.
2. Rosenmund, C., J. Rettig, and N. Brose, Molecular mechanisms of active zone function. *Curr Opin Neurobiol*, 2003. **13**(5): p. 509-19.

3. Richmond, J. and K. Broadie, The synaptic vesicle cycle: exocytosis and endocytosis in *Drosophila* and *C. elegans*. *Curr Opin Neurobiol*, 2002. **12**(5): p. 499.
4. Rettig, J. and E. Neher, Emerging roles of presynaptic proteins in Ca^{++} -triggered exocytosis. *Science*, 2002. **298**(5594): p. 781-5.
5. Sudhof, T.C., The synaptic vesicle cycle revisited. *Neuron*, 2000. **28**(2): p. 317-20.
6. Jahn, R. and T.C. Sudhof, Membrane fusion and exocytosis. *Annu Rev Biochem*, 1999. **68**: p. 863-911.
7. Fernandez-Chacon, R. and T.C. Sudhof, Genetics of synaptic vesicle function: toward the complete functional anatomy of an organelle. *Annu Rev Physiol*, 1999. **61**: p. 753-76.
8. Littleton, J.T. and H.J. Bellen, Presynaptic proteins involved in exocytosis in *Drosophila melanogaster*: a genetic analysis. *Invert Neurosci*, 1995. **1**(1): p. 3-13.
9. Sudhof, T.C., The synaptic vesicle cycle. *Annu Rev Neurosci*, 2004. **27**: p. 509-47.
10. Mezer, A., et al., A new platform to study the molecular mechanisms of exocytosis. *J Neurosci*, 2004. **24**(40): p. 8838-8846.
11. Gutman, M. and E. Nachliel, Time resolved dynamics of proton transfer in proteinous systems. *Annu. Rev. Phys. Chem.*, 1997. **48**: p. 329-56.
12. Berg, H.C. and E.M. Purcell, Physics of chemoreception. *Biophys J*, 1977. **20**(2): p. 193-219.
13. Mezer, A., et al., Systematic search for the rate constants that control the exocytotic process from chromaffin cells by a Genetic Algorithm. *BBA - Molecular Cell Research*, 2006: p. In press.
14. Voets, T., et al., Intracellular calcium dependence of large dense-core vesicle exocytosis in the absence of synaptotagmin I. *Proceedings of the National Academy of Sciences of the United States of America.*, 2001. **98**(20): p. 11680-5.
15. Voets, T., Dissection of Three Ca^{2+} -Dependent Steps Leading to Secretion in Chromaffin Cells from Mouse Adrenal Slices. *Neuron*, 2000. **28**(2): p. 537-545.

Author Index

- Ashery, Uri 249
- Bebek, Gürkan 119
- Benham, Craig J. 212
- Berenbrink, Petra 119
- Bosis, Eran 249
- Chen, Chih-Yu 138
- Chen, Su-Shing 1
- Chesler, Elissa J. 150
- Cooper, Colin 119
- de Hoon, Michiel 62
- Dill, David L. 11
- Fang, Jywe-Fei 166
- Friedetzky, Tom 119
- Gage, Pamela 11
- Guimarães, Katia S. 23
- Gutman, Menachem 249
- Hannenhalli, Sridhar 225
- Huang, Chien-Hung 166
- Husmeier, Dirk 188
- Ideker, Trey 39
- Jagalur, Manjunatha 95
- Jensen, Shane T. 225
- Jin, Shouguang 1
- Jonnalagadda, Sudhakar 178
- Jothi, Raja 23
- Kim, Haseong 70
- Knapp, Merrill A. 11
- Kulp, David 95
- Kuo, Chi-Li 138
- Laderoute, Keith 11
- Langston, Michael A. 150
- Lapidot, Michal 51
- Lee, Jae K. 70
- Lehrach, Wolfgang 188
- Lincoln, Patrick 11
- Man, Orna 107
- Mezer, Aviv 249
- Nachliel, Esther 249
- Nadeau, Joseph H. 119
- Ng, Ka-Lok 166
- Park, Taesung 70
- Pilpel, Yitzhak 51, 107
- Przytycka, Teresa M. 23
- Ruan, Jianhua 235
- Ruppin, Eytan 39
- Sahinalp, S. Cenk 119
- Sharan, Roded 39
- Shlomi, Tomer 39
- Song, Yongling 1
- Soo, Von-Wun 138
- Srinivasan, Rajagopalan 178
- Sussman, Joel L. 107
- Suthram, Silpa 39
- Sze, Sing-Hoi 198
- Talcott, Carolyn 11
- Tsai, Jeffrey J.P. 166
- Vitkup, Dennis 62
- Wang, Guandong 80, 235
- Wang, Huiquan 212
- Wang, Li-San 225
- Williams, Christopher K.I. 188
- Wu, Weihui 1
- Zhang, Weixiong 80, 235
- Zhao, Xiaoyan 198
- Zhou, Xuefeng 235
- Zotenko, Elena 23