Sackler Faculty of Exact Sciences, School of Computer Science

Computational analysis of molecular networks: modeling and reconstruction

THESIS SUBMITTED FOR THE DEGREE OF
"DOCTOR OF PHILOSOPHY"

by
Irit Gat-Viks

The work on this thesis has been carried out
under the supervision of **Prof. Ron Shamir**

Submitted to the Senate of Tel-Aviv University
January 2007

Acknowledgments

This thesis summarizes a wonderful period I spent in Tel-Aviv University. First and foremost, I would like to express my deepest gratitude to my advisor, Ron Shamir, for taking me under his wing and teaching me what science and research are all about. Ron allowed me to benefit from his ideas, knowledge, experience and support while not constraining my creative efforts. Ron, thank you for making me believe in myself.

I had a pleasure to work with many other gifted people. I am grateful to my collaborators: Amos Tanay, Roded Sharan, Richard M. Karp, Daniela Raijman and Igor Ulitsky. I want to thank Dan Graur for showing me the exciting world of molecular evolution and for being a friend. To Isaac Meilijson for teaching me everything I know about statistics, and for so much of his time and patience. Thanks to Martin Kupiec for being my genetics mentor during all my academic studies. Lots of thanks to all my lab friends: Einat Hazkani-Covo, Tal Dagan, Rani Elkon, Tzvika Hartman, Gadi Kimmel, Chaim Linhart, Adi Maron-Katz, Itsik Pe'er, Tal Pupko, Rotem Sorek, Israel Steinfeld and Michal Ziv-Ukelson. I deeply thank the Colton family for granting me a fellowship throughout my M.Sc. and Ph.D. studies.

Last but not least, I would like to thank my family. Thanks to my wonderful mother and beloved brother and sister for their unconditional love and support. Many thanks to my grandmother who initiated my interest in biology and taught me to appreciate and enjoy knowledge. I would like to thank my two most beloved daughters Netta and Hadas for granting me so much joy and happiness. This work is dedicated to my husband Amihai, without whom this thesis would not be possible. Thank you for being so loving and encouraging.

Preface

This thesis is based on the following collection of six articles that were published throughout the PhD period in scientific journals and in refereed proceedings of conferences.

1. **Scoring clustering solutions by their biological relevance.**
Irit Gat-Viks, Roded Sharan and Ron Shamir.
Published in *Bioinformatics* [1].

2. **Chain functions and scoring functions in genetic networks.**
Irit Gat-Viks and Ron Shamir.
Published in *Bioinformatics journal supplement for the proceedings of The 11th Annual International Conference on Intelligent Systems for MolecularBiology (ISMB 2003)* [2].

3. **Reconstructing chain functions in genetic networks.**
Irit Gat-Viks, Roded Sharan, Richard M. Karp and Ron Shamir.
Published in P*roceedings of the Pacific Symposium on Biocomputng (PSB 04)* [3] and in *SIAM journal of discrete mathematics* [4].

4. **Modeling and analysis of heterogeneous regulation in biological networks.**
Irit Gat-Viks, Amos Tanay and Ron Shamir.
Published in *Proceedings of the First RECOMB Satellite Workshop on Regulatory Genomics* [5] and in *Journal of Computational Biology (JCB)* [6].

5. **A probabilistic methodology for integrating knowledge and experiments on biological networks.**
Irit Gat-Viks, Amos Tanay, Daniella Raijman and Ron Shamir.
Published *in Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 05)* [7] and in *Journal of Computational Biology (JCB)* [8].

6. **Refinement and expansion of signaling pathways: the osmotic response network in yeast.**
Irit Gat-Viks and Ron Shamir.
To appear in *Genome Research* [9].

Abstract

A great challenge in understanding biological complexity is to reconstruct the molecular networks governing the activity within the cell. Recent high throughput techniques produce large scale measurements, which probe molecular networks from different perspectives. In this thesis, we describe our studies of molecular networks. We developed mathematical models for the representation of biological networks and provided algorithms for model reconstruction using large scale experimental data. Our computational methodologies accommodate information from a broad variety of sources and of diverse types, including general biological principles, established biological knowledge, and diverse large scale experiments. By using computational techniques from combinatorial optimization, probabilistic models and statistics, we could handle highly complex systems and large scale datasets. We used our methods on yeast, and showed that the integration of large scale data with mathematical modeling provides novel insights on the biological system and generation of hypotheses for further research.

Contents

Chapter 1

Introduction

1.1 The systems biology approach

High-throughput biotechnology enables the monitoring of thousands of biological molecules simultaneously. This allows a global view on the cellular activity under specific cellular conditions. The applications of such technology range from gene functional annotation and molecular network reconstruction to diagnosis of disease conditions and characterization of effects of medical treatments. Unlike the traditional biological approach of studying individual proteins or genes one at a time, the high throughput data make it possible to investigate thousands of molecular components simultaneously, and facilitate elucidation of global regulatory principles. Thus, for the first time in history, it is possible to obtain a comprehensive understanding of the tremendous complexity of life.

Since the generation of high-throughput data has been greatly accelerated, major efforts are invested in developing computational methodologies to analyze and extract information from the data. The high-throughput data are integrated, visualized, and modeled computationally. To date, despite the intensive studies and even in well studied organisms, signaling pathways and regulation of gene expression are still far from being completely understood, and many proteins are still uncharacterized.

To improve our understanding, we need to build computational models and analyze them. Computational models can provide different kinds of insights: (i) **General properties** - Global analysis of a model may provide understanding of general properties, such as the scale free property in network models [10], and common model substructures called network motifs [11, 12]. (ii) **Functional annotation** - Protein function can be elucidated by interpretation of the data in the context of a computational model. For example, network models are used to propagate functional annotation from one protein to another [13]. Alternatively, functional modules are used for annotation based on majority rules [14]. (iii) **System behavior** - mathematical models can generate predictions of system state(s) under different conditions [15-17]. In this thesis, we focus on the latter

type of questions, aiming to design and analyze predictive models for molecular networks.

1.2 Cellular activity can be measured in a genomic scale

High throughput gene expression measurements [18, 19] are currently the most popular kind of functional genomic information. The technology of microarrays (DNA chips) allows the measurement of thousands of mRNAs molecules simultaneously. The resulting gene expression profiling is now a standard tool in many biological laboratories, with applications to functional annotation, tissue classification, and regulatory motif identification [20-22].

Many other high throughput techniques for probing biological systems are constantly emerging. The information obtained from these techniques can be classified into three categories: Information about the abundance of molecular components, about the function of molecular components, and about interaction between components. The abundance is measured for mRNA molecules (as mentioned above), protein molecules (Mass spectrometry technology, see [23]) and metabolites (NMR and vibrational spectrometry, see [24]). Data concerning the interactions between protein and DNA is measured by ChIP-chip technology (using whole genome promoters [25] or tiling arrays [26]) and provide information on the location of transcription factors on their target promoters. Protein-Protein interactions provide information on signaling cascades and protein complexes based on two-hybrid systems [27] or mass spectrometry. Functional information is available using various experimental techniques such as synthetic-lethal interactions [28], single-gene deletion microarray data [29], and global kinase effects [30, 31].

Interpretation of the high-throughput data sets is not easy and straightforward. The data sets contain experimental noise, missing information and many technical artefacts and biases. Many datasets are immense in size and have no agreed upon, standard representation. Most importantly, each experimental technique measures only one type of information, which can at best be used as a rough approximation for other types of information needed. Despite the many problems with this data, researchers are making progress by modeling and integrating together different kinds of genome-wide data sets.

1.3 Modeling approaches

We can distinguish between three different levels of increasing detail in computational models:

(i) **Functional groups model** - This classical modeling approach dissects the molecular components (e.g., genes, proteins) into groups with a common functionality. The groups are called *functional groups*, *parts lists*, *gene/protein sets*, *clusters,* or *modules*. This approach provides a rough understanding of the global architecture of the system, and at the same time, aims to derive highly specific predictions on the components' functionality.

(ii) **Topology model** (interactions network) - This modeling approach describes the structural features of the network. The model is a wiring diagram (graph), where nodes represent molecular components and edges represent interactions among them. The common models are *protein-protein interaction networks* and *transcriptional gene regulatory networks.*

(iii) **Predictive model** - The heterogonous high-throughput information makes it possible to go beyond modules and topology, and reconstruct a logical model. The model describes how the interactions give rise to the function and behaviour of the system, and represents either steady state or real-time dynamic behavior of the system. The model computes the expected response to various external or internal stimuli, and allows simulation of cellular behavior.

The preparation of computational models requires reliable high throughput data. However, the current technology provides noisy and biased information. In addition, biological systems are complex and have highly intricate regulatory mechanisms. Such complex systems cannot be fully characterized based on a single experimental technique. Hence, in order to obtain reliable models, the reconstruction algorithms should integrate multiple independent data sets, thereby supporting the conclusions through several independent types of information. In the following sections we discuss each of the modeling approaches and show how it can be used to integrate multiple data sets.

1.3.1 Clustering algorithms identify functional groups

A central step in the analysis of whole-genome mRNA (or protein) abundance is the identification of groups of genes (proteins) that exhibit similar expression patterns. This

translates to the algorithmic problem of clustering. In a clustering problem, the goal is to partition the elements into subsets, called clusters, so that two criteria are satisfied: *Homogeneity*- elements in the same cluster are similar to each other, and *separation*- elements from different clusters are dissimilar. Clustering methods are used to partition a very large matrix of expression levels to more informative subsets of gene or conditions, which are assumed to share functionality or to form some biological modules. Many standard clustering algorithms are commonly used, including hierarchical clustering [22], k-means, and self organizing maps [32]. Other algorithms are specific and take into consideration the special properties of the high throughput biological data [33, 34]. The clustering approach was shown to be instrumental in functional annotation, tissue classification, motif identification, operon prediction and more [20-22, 35-37]. Biclustering methods, which seek a subset of genes that exhibit a similar behavior across a subset of conditions, were also proved to be useful in expression data analysis [38-40]. Usually, given a gene set of functionally-related genes or proteins, their common functionality is identified by a functional enrichment, or enrichment of transcription factors binding motifs [14, 41].

Instead of clustering new genome-wide experiments separately from any other data sets, more robust sets can be constructed by integrating multiple datasets together (e.g., [42, 43]). Alternatively, recent methods used gene sets as basic building blocks for additional analyses [44, 45]. Hence, instead of having to understand the behavior of thousands of individual components, one can focus on a much smaller set of clusters. Moreover, by considering the joint behavior of a large set, we can detect delicate changes that are not significantly identified on individual genes.

1.3.2 Interaction networks represent the backbone of molecular activity

A. Transcriptional networks

Although clustering of gene expression data is a useful way to identify groups of genes that are involved in a complex coordinated activity, it only tells us which genes are co-regulated, without describing the regulatory rules. Discovering these rules requires reconstruction of the causal interactions between transcription factors (TFs) and gene targets, i.e., the transcriptional network. The most straightforward way to build the network is by using DNA-protein binding indicated by ChIP-on-chip results. These data

must be integrated with other data sources, since it is still prone to high error rate. The prevalent integration paradigms are the following:

(i) **Sequence analysis of a single or multiple genomes** - Transcription factor binding sites can help in exposing the transcriptional network. The binding sites can be revealed based on their abundance in a single genome (for example, the MEME algorithm [46]), or based on their pattern of evolution in the genomes of a few closely related organisms [47-49].

(ii) **Combine genome sequence and expression data** - In the basic method, after identifying a group of co-regulated genes based on gene expression data, the regulating TFs are revealed by overrepresentation of their cis-regulatory motifs [14]. Other methods find the motifs without the need for clustering [50, 51].

(iii) **Combine DNA-protein binding data and expression data -** The basic approach is to cluster genes that share the same expression pattern and then identify their TFs using enrichment of DNA-protein binding data in their promoters. More sophisticated approaches revise the clustering solution according to the binding data, in order to obtain gene sets to which the same combination of TFs bind (e.g., [52, 53]).

B. Protein-protein networks

The modeling of the protein-protein interaction network requires reliable interaction data sets, but current data are noisy and may be misleading. Hence, researchers are trying to predict interactions using data integration from two or more distinct protein-protein interactions assays, as well as from genome sequence, functional knowledge, co-localization, and homologue interactions in other species [54-56].

Although the current interaction networks are incomplete, they still provide a useful framework for extracting biological insights. First, it is possible to generate new global hypotheses based on the interaction network [10-12, 57]. Second, it is possible to improve the functional annotation by employing "guilt by association" and majority rule principles in the context of the network (reviewed in [13]).

1.3.3 Predictive models represent the behavior of the system

Traditionally, molecular biology was an informal data-poor science, which applied a reductionist hypothesis-driven approach. Informal static flowchart-type models of specific pathways were used as a basic tool, providing a mental picture of the biological

system. The flowcharts conveyed information easily, and enabled researchers to understand experimental results and plan experiments. However, given high-throughput information, it is impossible to compare the charts to the huge amount of data manually. Moreover, the large amount of available knowledge makes it impossible to mentally capture and manage biological systems. In this situation, the informal flowcharts should be replaced by formal mathematical predictive models, which are analyzed systematically by computational algorithms.

A *predictive model* is a simplifying formal abstraction of the biological system, which generates predictions of the system behavior under different conditions. Typically, these models are represented by a network with an underlying logic. The nodes are biological components, and the *structure* (topology) represents regulatory relations among components. Each of the nodes is associated with a *regulation function*, which describes its logic of regulation. Hence, the function represents the content (or production rate) of the component given the content of its upstream components. The structure tells "who acts on whom" and the functions tell how. In biological context, the mathematical models are referred to as *cellular networks* and *molecular networks*, and sometimes more specifically as *signaling network* or *metabolic network*. Standard computational models and learning algorithms can be applied to construct predictive mathematical models of biological systems.

A large variety of possible model types are used in the biological and bioinformatics communities. Modeling decisions include discrete vs. continuous, static vs. dynamic, deterministic vs. probabilistic, and various levels of detail. Perhaps the most basic model types are Boolean [58-60], qualitative [61], linear [62], differential equations [63], and Bayesian networks [64-66].

In all levels of resolution, a key obstacle in trying to reconstruct a mathematical predictive model from data is a large solution space: There are too many possible solutions, and consequently an unrealistic amount of data is needed to identify the right one. Due to this inherent complexity of the network, the solution space must be limited using prior biological knowledge. The prior information can be classified as *qualitative* vs. *quantitative*, *structural* (i.e., information about topology) vs. *mechanistic* (i.e., information about regulation functions), and *general* vs. *specific* knowledge on particular components. Moreover, the prior knowledge can be used to bias the solution (i.e., used as a soft prior) or to impose constraints on the solution. The prior information used in the current literature for model reconstruction can be organized in four categories:

1. Qualitative constraints on the structure. Several works impose general biological constraints based on general knowledge and understanding of biological networks. For example, Gardner et al. [67] proposed a linear model of the SOS response in *E. coli* assuming maximum of k regulatory inputs. Segal et al. [68] used a Bayesian network model assuming decomposition of the network into modules of co-regulated genes.

2. Quantitative constraints on the structure. Independent high-throughput datasets are commonly used as soft priors. For example, Imoto et al. [69] used protein-protein interactions, protein-DNA interactions, and binding site information as priors for Bayesian networks reconstruction. Herrgard et al. [70] used ChIP-on-chip results as prior for elucidating the set of transcriptional target genes based on gene expression data.

3. Qualitative knowledge on reaction mechanisms. In most of the cases, even if we understand the mechanism quite well, the exact parameters are unknown and the prior is only qualitative. The 'physicochemical approach' uses the available qualitative physical or chemical knowledge to formalize a differential equations model. The missing kinetic parameters are filled by a calibration process (reviewed in [15], see for example [71-73]). Alternatively, a probabilistic factor graph model [74] has been proposed to model known interactions, their directionality and the sign of the immediate effects [75].

4. Quantitative knowledge on reaction mechanisms. Measured kinetic rate constants and stoichiometic coefficients are used as parameters in physicochemical modeling and in metabolic engineering [15, 76].

Mathematical modeling of diseases is an essential step in the development of new drugs, medical diagnostics and therapies. Up to now, most medical computational research has focused on cluster analysis, statistical tests of hypotheses, and classification of diseases subtypes. However, mathematical modeling may contribute to the understanding of disease processes, provide quantitatively clinically relevant parameters, simulate genetic diseases, and predict drug responses. Hence, mathematical models carry the promise to become an important tool in personalized medicine and pharmacological research [77].

1.4 Summary of articles included in this thesis

1. Scoring clustering solutions by their biological relevance
Irit Gat-Viks, Roded Sharan and Ron Shamir.
Published in *Bioinformatics* [1].

A central step in the analysis of gene expression data is the identification of groups of genes that exhibit similar expression patterns. Clustering gene expression data into homogeneous groups was shown to be instrumental in functional annotation, tissue classification, regulatory motif identification, and other applications. Although there is a rich literature on clustering algorithms for gene expression analysis, very few works addressed the systematic comparison and evaluation of clustering results. Typically, different clustering algorithms yield different clustering solutions on the same data, and there is no agreed upon guideline for choosing among them.

We developed a novel statistically based method for assessing a clustering solution according to prior biological knowledge. Our method can be used to compare different clustering solutions or to optimize the parameters of a clustering algorithm. The method is based on projecting vectors of biological attributes of the clustered elements onto the real line, such that the ratio of between-groups and within-group variance estimators is maximized. The projected data are then scored using a non-parametric analysis of variance test, and the score's confidence is evaluated. We validate our approach using simulated data and show that our scoring method outperforms several extant methods, including the separation to homogeneity ratio and the silhouette measure. We apply our method to evaluate results of several clustering methods on yeast cell-cycle gene expression data.

2. **Chain functions and scoring functions in genetic networks**
Irit Gat-Viks and Ron Shamir.

Published in *Bioinformatics journal supplement for the proceedings of The 11th Annual International Conference on Intelligent Systems for MolecularBiology (ISMB 2003)* [2].

One of the grand challenges of system biology is to reconstruct the network of regulatory control among genes and proteins. High throughput data, particularly from expression experiments, may gradually make this possible in the future. Here we address two key ingredients in any such 'reverse engineering' effort: The choice of a biologically

relevant, yet restricted, set of potential regulation functions, and the appropriate score to evaluate candidate regulatory relations. We propose a set of regulation functions which we call chain functions, and argue for their ubiquity in biological networks. We analyze their complexity and show that their number is exponentially smaller than all boolean functions of the same dimension. We define two new scores: one evaluating the fitness of a candidate set of regulators of a particular gene, and the other evaluating a candidate function. Both scores use established statistical methods. Finally, we test our methods on experimental gene expression data from the yeast galactose pathway. We show the utility of using chain functions and the improved inference using our scores in comparison to several extant scores. We demonstrate that the combined use of the two scores gives an extra advantage. We expect both chain functions and the new scores to be helpful in future attempts to infer regulatory networks.

3. Reconstructing chain functions in genetic networks

Irit Gat-Viks, Roded Sharan, Richard M. Karp and Ron Shamir.

Published in P*roceedings of the Pacific Symposium on Biocomputing (PSB 04)* [3]. A journal version was published in *SIAM journal of discrete mathematics [4]*.

This study builds on the chain function paradigm introduced in the previous study. In this paper we study the computational problem of reconstructing a chain function using a minimum number of experiments, in each of which only few genes are perturbed. We address both the question of finding the set of regulators of a chain function, which is typically much smaller than the entire set of genes, and the question of reconstructing the function given its regulators. We give optimal reconstruction schemes for several scenarios and show their application on real data. Our analysis focuses on the theoretical complexity of reconstructing regulation relations, assuming that experiments provide accurate results and that the target function can be studied in isolation from the rest of the genetic network.

4. Modeling and analysis of heterogeneous regulation in biological networks

Irit Gat-Viks, Amos Tanay and Ron Shamir.

Published in *Proceedings of the First RECOMB Satellite Workshop on Regulatory Genomics* and *in Journal of Computational Biology (JCB)*[6].

In this study, we propose a novel model for the representation of biological networks and provide algorithms for learning model parameters from experimental data.

Our approach is to build an initial model based on extant biological knowledge and refine it to increase the consistency between model predictions and experimental data. Our model encompasses networks that contain heterogeneous biological entities (mRNA, proteins, metabolites) and aims to capture diverse regulatory circuitry on several levels (metabolism, transcription, translation, post-translation and feedback loops, among them). Algorithmically, the study raises two basic questions: how to use the model for predictions and inference of hidden variables states, and how to extend and rectify model components. We show that these problems are hard in the biologically relevant case where the network contains cycles. We provide a prediction methodology in the presence of cycles and a polynomial time, constant factor approximation for learning the regulation of a single entity.

A key feature of our approach is the ability to utilize both high-throughput experimental data, which measure many model entities in a single experiment, as well as specific experimental measurements of few entities or even a single one. In particular, we use together gene expression, growth phenotypes, and proteomics data. We tested our strategy on the lysine biosynthesis pathway in yeast. We constructed a model of more than 150 variables based on an extensive literature survey and evaluated it with diverse experimental data. We used our learning algorithms to propose novel regulatory hypotheses in several cases where the literature-based model was inconsistent with the experiments. We showed that our approach has better accuracy than extant methods of learning regulation.

5. A probabilistic methodology for integrating knowledge and experiments on biological networks

Irit Gat-Viks, Amos Tanay, Daniella Raijman and Ron Shamir.

Published *in Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 05)* [7] and in *Journal of Computational Biology (JCB)* [8].

This study generalizes the model introduced in the previous work from combinatorial to probabilistic setting. Here we introduce an extended computational framework that combines formalization of existing qualitative models, probabilistic modeling, and integration of high-throughput experimental data. Using our methods, it is possible to interpret genome-wide measurements in the context of prior knowledge on the system, to assign statistical meaning to the accuracy of such knowledge, and to learn refined models with improved fit to the experiments. Our model is represented as a

probabilistic factor graph, and the framework accommodates partial measurements of diverse biological elements. We study the performance of several probabilistic inference algorithms and show that hidden model variables can be reliably inferred even in the presence of feedback loops and complex logic. We show how to refine prior knowledge on combinatorial regulatory relations using hypothesis testing and derive p-values for learned model features. We test our methodology and algorithms on a simulated model and on two real yeast models. In particular, we use our method to explore uncharacterized relations among regulators in the yeast response to hyper-osmotic shock and in the yeast lysine biosynthesis system. Our integrative approach to the analysis of biological regulation is demonstrated to synergistically combine qualitative and quantitative evidence into concrete biological predictions.

6. **Refinement and expansion of signaling pathways: the osmotic response network in yeast**

Irit Gat-Viks and Ron Shamir.

Published in *Genome Research* [9].

In this study we continue the development of the modeling and analysis strategy introduced in the previous paper. We present algorithms that analyze experimental results (e.g., transcription profiles) vis-à-vis the model, and propose improvements to the model based on the fit to the experimental data. These algorithms refine the relations between model components, as well as expand the model to include new components that are regulated by components of the original network.

Using our methodology, we have modeled together the knowledge on four established signaling pathways related to osmotic shock response in S. cerevisiae. Using over 100 published transcription profiles, our refinement methodology revealed three cross-talks in the network. The expansion procedure identified with high confidence large groups of genes that are co-regulated by transcription factors from the original network via a common logic. The results reveal novel delicate repressive effect of the HOG pathway on many transcriptional target genes, and suggest an unexpected alternative functional mode of the MAP kinase Hog1. The analysis also predicts novel feed-forward and feedback loops in the regulatory network, which probably support cellular adaptation to osmotic stress. These results demonstrate that by integrated analysis of data and of well-defined knowledge on signaling pathways, one can generate concrete biological hypotheses about signaling cascades and their downstream regulatory programs.

Chapter 2

Articles

BIOINFORMATICS

Vol. 19 no. 18 2003, pages 2381–2389
DOI: 10.1093/bioinformatics/btg330

Scoring clustering solutions by their biological relevance

I. Gat-Viks[1,*,†], R. Sharan[2,†] and R. Shamir[1]

[1]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel and
[2]International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704, USA

Received on February 5, 2003; revised on May 20, 2003; accepted on June 3, 2003

ABSTRACT

Motivation: A central step in the analysis of gene expression data is the identification of groups of genes that exhibit similar expression patterns. Clustering gene expression data into homogeneous groups was shown to be instrumental in functional annotation, tissue classification, regulatory motif identification, and other applications. Although there is a rich literature on clustering algorithms for gene expression analysis, very few works addressed the systematic comparison and evaluation of clustering results. Typically, different clustering algorithms yield different clustering solutions on the same data, and there is no agreed upon guideline for choosing among them.

Results: We developed a novel statistically based method for assessing a clustering solution according to prior biological knowledge. Our method can be used to compare different clustering solutions or to optimize the parameters of a clustering algorithm. The method is based on projecting vectors of biological attributes of the clustered elements onto the real line, such that the ratio of between-groups and within-group variance estimators is maximized. The projected data are then scored using a non-parametric analysis of variance test, and the score's confidence is evaluated. We validate our approach using simulated data and show that our scoring method outperforms several extant methods, including the separation to homogeneity ratio and the silhouette measure. We apply our method to evaluate results of several clustering methods on yeast cell-cycle gene expression data.

Availability: The software is available from the authors upon request.

Contact: iritg@post.tau.ac.il; rshamir@post.tau.ac.il; roded@icsi.berkeley.edu

INTRODUCTION

DNA microarray technology enables the monitoring of expression levels of thousands of genes simultaneously. This allows a global view on the transcription levels of many genes under specific cellular conditions. The applications of such technology range from gene functional annotation and genetic network reconstruction to diagnosis of disease conditions and characterization of effects of medical treatments.

A central step in the analysis of gene expression data is the identification of groups of genes that exhibit similar expression patterns. Clustering methods transform a large matrix of expression levels into a more informative collection of gene sets (or condition sets) which are assumed to share biological properties. Clustering gene expression data into homogeneous groups was shown to be instrumental in functional annotation, tissue classification, motif identification, and other applications [for a review see Sharan *et al.* (2002)].

Although there has been extensive research on clustering algorithms for gene expression analysis (Eisen *et al.*, 1998; Tamayo *et al.*, 1999; Ben-Dor *et al.*, 1999; Sharan and Shamir, 2000; Sharan *et al.*, 2003), very few works have been published on the systematic comparison and evaluation of clustering results. Typically, different clustering algorithms yield different clustering solutions on the same data, and often the same algorithm yields different results for different parameter settings, and there is no consensus on choosing among them.

Different measures for the quality of a clustering solution are applicable in different situations, depending on the data and on the availability of the true solution. In case the true solution is known, and we wish to compare it to another solution, one can use, e.g. the Minkowski measure (Sokal, 1977) or the Jaccard coefficient [cf. Everitt (1993)]. When the true solution is not known, there is no agreed-upon approach for evaluating the quality of a suggested solution. Several approaches evaluate a clustering solutions based on its intra-cluster homogeneity or inter-cluster separation (Hansen and Jaumard, 1997; Sharan *et al.*, 2003; Yeung *et al.*, 2001). However, the homogeneity and separation criteria are inherently conflicting, as an improvement in one will usually correspond to worsening of the other. One way of getting around this problem is to fix the number of clusters and seek a solution with maximum homogeneity. This is done, for example, by the classical *K*-means algorithm (MacQueen, 1965; Ball and Hall,

*To whom correspondence should be addressed.

†These authors contributed equally to this work.

1967). For methods that evaluate the number of clusters see, e.g. Hartigan (1975); Tibshirani *et al.* (2000); Ben-Hur *et al.* (2002); Pollard and van der Laan (2002); Dudoit and Fridlyand (2002); McLachlan (1987). Another way to overcome the problem is by presenting a curve of homogeneity versus separation (Ben-Dor, private communication). Such a curve can show that one algorithm dominates another if it provides better homogeneity for all separation values, but typically different algorithms will dominate in different value range. An alternative method suggested by Kaufman and Rousseeuw (1990), evaluates a solution using a numerical measure called the average silhouette. This method performs well in general, but fails to detect fine cluster structures (Pollard and van der Laan, 2002).

Clustering quality can also be visually assessed by using discriminant analysis [e.g. Stephanopoulos *et al.* (2002); McLachlan (1992)] or principal component analysis [e.g. Mendez *et al.* (2002)], that reduce data dimensionality. Single clusters can be scored based on prior biological knowledge, e.g. by checking for functional enrichment of genes in a cluster or searching for common motifs in their promoter regions (Tavazoie *et al.*, 1999). Clustering solutions can in some cases be assessed by applying standard statistical techniques. For high-dimensional data, multivariate analysis of variance (MANOVA) and discriminant analysis (Huberty, 1994; Mendez *et al.*, 2002) are appropriate if the data are normally distributed. For the case of non-normal data, there are several extensions that require the data to be either low-dimensional (Bishop *et al.*, 1975) or continuous (Katz and McSweeney, 1980). If attributes are independent one can also test the significance of the grouping for each dimension separately, and combine the resulting scores (Pesarin, 2001). None of these methods apply when wishing to test the significance of a clustering solution based on high-dimensional vectors of dependent biological attributes that do not necessarily follow a normal distribution and may even be discrete.

In this paper we devise a statistically based method for comparing clustering solutions according to prior biological knowledge. In our method, solutions are ranked according to their correspondence to prior knowledge about the clustered elements. Given a vector of (continuous or discrete) attributes for each element, our method tests the dependency between the attributes and the grouping of the elements. The test is applied simultaneously to all the attributes. In our application, elements are genes, clustered according to their expression patterns, and the attributes of a gene are binary indicators of its membership in specific functional classes. In this case, the method computes a quality score for the functional enrichment of these classes among each solution's clusters. At the heart of our method is a projection of the high-dimensional data to one dimension, to avoid the problem of applying MANOVA to the data. Using the one-dimensional data, the solutions are compared based on their score in a non-parametric ANOVA test.

In the rest of the paper, after providing some background, we describe our method, and give results on its performance on simulated and real data.

PRELIMINARIES

The input to a clustering problem consists of a set of elements and a characteristic vector for each element. A measure of (dis)similarity is defined between pairs of such vectors. (In gene expression, elements are usually genes, and the vector of each gene contains its expression levels under each of the monitored conditions. Dissimilarity between vectors can be measured, e.g. by their Euclidean distance.) The goal is to partition the elements into subsets, which are called *clusters*, so that two criteria are satisfied: homogeneity – elements in the same cluster are similar to each other; and separation – elements from different clusters are dissimilar.

Let N be a set of n elements and let $\mathcal{C} = \{C_1, \ldots, C_l\}$ be a partition of these elements into l clusters. We call two elements from the same cluster *mates* (with respect to $\mathcal{C}$). A common procedure for evaluating a clustering solution given the true solution, is to compute its *Jaccard coefficient* [see, e.g. Everitt (1993)], which is the proportion of correctly identified mates out of the sum of the correctly identified mates plus the total number of disagreements (pairs of elements that are mates in exactly one of the two solutions). Hence, a perfect solution has score 1, and the higher the score – the better the solution. When the true solution is not known, a solution can be evaluated by its homogeneity and separation. The *homogeneity* of $\mathcal{C}$ is the average distance between mates, and the *separation* of $\mathcal{C}$ is the average distance between non-mates (Hansen and Jaumard, 1997; Sharan *et al.*, 2003). Another popular measure is the average silhouette (Kaufman and Rousseeuw, 1990), which is computed as follows: define the *silhouette* of element j as $(b_j - a_j)/\max(a_j, b_j)$, where a_j is the average distance of element j from other elements of its cluster, b_{jk} is the average distance of element j from the members of cluster C_k, and $b_j = \min_{\{k: j \notin C_k\}} b_{jk}$. The *average silhouette* is the mean of this ratio over all elements.

Our main focus is the evaluation of clustering solutions using external information. The setup for the problem is as follows: we are given an $n \times p$ *attribute matrix* A. The rows of A correspond to elements, and the ith row vector is called the *attribute vector* of element i. We are also given a clustering $\mathcal{C} = \{C_1, \ldots, C_l\}$ of the elements, where $s_i = |C_i|$. For convenience, we shall also index the attribute vectors by the clustering, i.e. use $a_{ij} = (a_{ij}^1, \ldots, a_{ij}^p)$ as the vector of element j in cluster i. Typically $\mathcal{C}$ is obtained without using the information in A. Our goal is to evaluate $\mathcal{C}$ with respect to A.

When $p = 1$, there are established statistical tests for the problem. Such tests will serve as building blocks in our method. In the case that the attribute is normally distributed, and under the assumption that the variances of the l population distributions are identical, we can use standard

analysis of variance (ANOVA) methods to test the significance of the grouping [see, e.g. Sokal and Rohlf (1995)]: suppose that the attribute of element j in cluster i has value a_{ij}. Let $\bar{a}_i$ denote the mean of the elements in cluster i, and let $\bar{a}$ denote the total mean of all n elements. When ANOVA is carried out, the null hypothesis is that the groups do not differ in location, i.e. $H_0 : \mu_1 = \mu_2 = \cdots = \mu_l$, where μ_i is the expectation of group i. The test statistic typically used is the ratio of variance estimator, i.e. the ratio of the hypothesis (or between-groups) mean square (MSH) to the error mean square (MSE):

$$F_H = \frac{\text{MSH}}{\text{MSE}} = \frac{\text{SSH}/(l-1)}{\text{SSE}/(n-l)} \tag{1}$$

where the hypothesis sum of squares is $\text{SSH} = \sum_{i=1}^{l} s_i(\bar{a}_i - \bar{a})^2$ and the error sum of squares is $\text{SSE} = \sum_{i=1}^{l}\sum_{j=1}^{s_i}(a_{ij} - \bar{a}_i)^2$. Under certain data conditions the F_{H} statistic has a (central) F distribution with $l-1$ and $n-l$ degrees of freedom.

In case the attribute (or some transformation of it) does not follow a normal distribution, one can use the Kruskal–Wallis (KW) test [cf. Sokal and Rohlf (1995)] as a non-parametric ANOVA test. The test assumes that the clusters are independent and have similar shape. We shall denote by $P^{\text{KW}}(\mathcal{C}, A)$ the p-value obtained by the KW test for a clustering $\mathcal{C}$ using the attribute $A : N \to R$. For the multidimensional case ($p > 1$), the MANOVA test [cf. Sokal and Rohlf (1995)] applies the same objective function F_{H}, but it applies only if the attribute matrix is multinormally distributed.

METHOD

Our goal is to evaluate a clustering solution given an attribute vector for each element, which represents the prior biological knowledge about the element. To this end, the MANOVA test is particularly appealing, as the numerator in Equation (1) (MSH) measures the separation (normalized by the number of clusters) and the denominator (MSE) measures the (normalized) homogeneity. However, the distribution of attribute vectors does not necessarily meet the requirements of MANOVA test. Such is the case, in particular, when attributes are binary. Thus, we propose to project the high-dimensional attribute vectors onto the real line using a linear combination of the attributes. Then, the solution $\mathcal{C}$ is scored by a non-parametric one-way ANOVA test on the one-dimensional data. We refer to the result as the *CQS* (Clustering Quality Score) of the clustering. CQS is computed as follows:

1. *Computing a linear combination of the attributes*. Each element is assigned a real value, which is a weighted sum of its attributes. An attribute's *weight* is its coefficient in the linear combination. Intuitively, we would like to weight the attributes such that they will contribute to the solution score according to their 'importance'. Usually, we do not know in advance the desired weighting of the attributes. In such cases, we propose to use weights that maximize the ability to discriminate between the clusters using the one-dimensional data. Finding the weights will be done in the same manner as in Linear Discriminant Analysis (LDA) (Huberty, 1994). The procedure for weight finding does not require any assumptions on the distribution of A. LDA creates such a linear combination by maximizing the ratio of between-groups-variance to within-groups-variance, as follows: let w be some p-dimensional vector of weights. The statistic being maximized is the ratio of MSH to MSE:

$$F(w) = \frac{\sum_{i=1}^{l} s_i(w \cdot \bar{a}_i - a \cdot \bar{a})^2/(l-1)}{\sum_{i=1}^{l}\sum_{j=1}^{s_i}(w \cdot a_{ij} - w \cdot \bar{a}_i)^2/(n-l)} \tag{2}$$

where $\bar{a}_i$ is the mean vector of cluster i, and $\bar{a}$ is the total mean vector. When introducing an additional constraint of a unit denominator, the maximum value of $F(w)$ is proportional to the greatest root of the equation $|H - \lambda E| = 0$. Here, H is a $p \times p$ matrix containing the between-groups sum of square $H_{rs} = \sum_{i=1}^{l} s_i(\bar{a}_i^r - \bar{a}^r)(\bar{a}_i^s - \bar{a}^s)$, and E is a $p \times p$ matrix of the sum of squared errors $E_{rs} = \sum_{i=1}^{l}\sum_{j=1}^{s_i}(a_{ij}^r - \bar{a}_i^r)(a_{ij}^s - \bar{a}_i^s)$, where $\bar{a}_i^r$ is the mean of attribute r in cluster i and $\bar{a}^r$ is the total mean of attribute r. Thus, the desired combination w is the eigenvector corresponding to the greatest root. This result holds without assuming any prior distribution on the attributes.

2. *Projection*. Apply the linear combination w to the attribute vectors, thereby projecting these vectors onto the real line. That is, $z_{ij} = \sum_t a_{ij}^t w^t$.

3. *Computing CQS using the projected values*. We now evaluate the clustering vis-á-vis the projected attributes using the KW test. We define CQS as $-\log p$, where $p = P^{\text{KW}}(\mathcal{C}, Z)$, i.e. the p-value assigned to the clustering by the KW test. Note that p is not the probability of observing the original attributes allocation randomly, since the vector data was first projected to maximize the variance ratio. Rather, the p-value is the probability that all values in this particular projection have been taken from the same population. Hence, CQS favors clustering solutions whose best discriminating weights enable significant grouping.

4. *Estimating confidence*. In order to estimate the accuracy of the scores and the significance of differences between the scores of distinct solutions, we evaluate the sensitivity of CQS to small modifications of the clustering solution. Intuitively, the larger the influence of small perturbations in the clustering on the CQS value, the smaller the confidence we have in the CQS. Specifically, for a given original solution we generate a group of alternative clustering solutions. Each alternative solution is obtained by introducing k exchanges of random pairs of elements from different clusters of the original solution (k is typically small, such as 2% of the elements). The *CQS confidence* is the standard deviation of CQS for the group of alternative clustering solutions.

The overall procedure is as follows:

1. Find the eigenvector w corresponding to the greatest root of the system of equations $|H - \lambda E| = 0$.
2. For each attribute vector a_{ij} set $z_{ij} = \sum_t a_{ij}^t w^t$.
3. Compute $p = P^{\mathrm{KW}}(\mathcal{C}, Z)$; let $\mathrm{CQS}(\mathcal{C}, A) = -\log p$.
4. Estimate the statistical confidence of the result by perturbations on $\mathcal{C}$.

Our scoring scheme can be applied in several ways and for several purposes. Our focus in this study is the evaluation of clustering solutions given external biological attributes, that were not used in the clustering process. Another application of our score is internal validation of solutions based on the same attributes that were used in generating the clustering. This can help in choosing among different clustering algorithms, as well as in optimizing the parameters of a specific algorithm (for example, choosing the number of clusters for K-means).

RESULTS

The score calculation was implemented in Perl under linux, using MATLAB. Running time for a data set of 750 clustered elements and 80 attributes is about a minute, on a standard 800 MHz PC. Below we report on the performance of our method on simulated and real data.

Simulations

We validate our method by conducting a series of tests on simulated data. We tested the effect of the one-dimensional projection of the attribute vector, the sensitivity of CQS to the solution accuracy, and the ability of CQS to pinpoint the right number of clusters and to detect fine clustering structures.

The data were generated as follows: profiles of 80 binary attributes were generated for five groups of $n = 50$ genes each. (We use the term 'genes' for uniformity. The simulations test the score irrespective of the nature of the clustered elements.) For each attribute we randomly selected one group in which its frequency will be r, and in the other four groups its frequency was set to r_0. The set of r (r_0) genes with that attribute was randomly selected from the relevant groups. $r_0 = 5$ was used throughout. Since we randomly select for each attribute the single group with frequency r, the overall density of the attribute vectors should be about the same for all elements, and the distinction must be based on individual attributes. Clearly, the larger the difference between r and r_0, the easier the distinction between the groups.

A. *The effect of one-dimensional projection*. First, we wished to examine the effect of reducing the attribute dimension to 1. We simulated data sets with $r = 6, 10, 15, 20$ and 25. For each data set we computed the ratio of separation to homogeneity of the true clustering on the original data (S/H) and on the projected data (S^*/H^*). This procedure was repeated 10 times. The results are shown in Figure 1B. Clearly, the

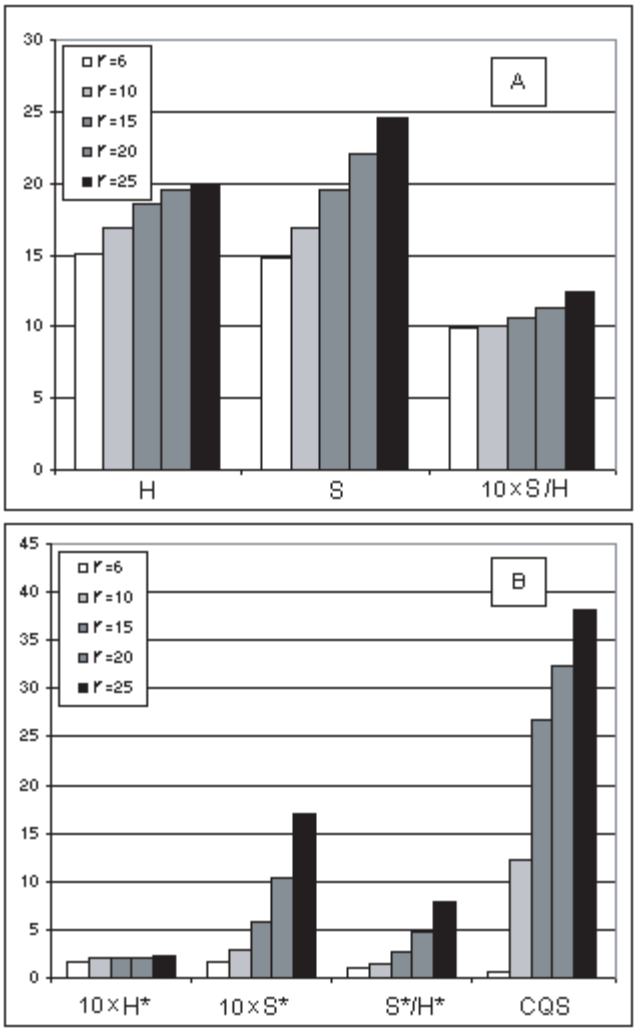


Fig. 1. Clustering parameters on simulated data. Y-axis: scores of five simulation setups $r = 6, 10, 15, 20, 25$ in different gray scale colors. (**A**) Scores are Homogeneity (H), separation (S) and their ratio on the original data. (**B**) Scores are Homogeneity (H^*), separation (S^*), their ratio and CQS on the projected (reduced) data. Numbers are average of 10 runs.

monotonicity of the homogeneity, separation and their ratio as a function of r, which is manifested on the original data, is preserved on the reduced data. The same monotonicity was observed in each of the 10 repetitions. Also, as expected, CQS improves monotonically with r.

The projected data for two simulations with $r = 6$ and $r = 25$ are visualized in Figure 2. For $r = 6$, the clusters look very similar, even though the data were reduced using the best separating linear combination. On the other hand, for $r = 25$, inter-cluster separation of most clusters is clearly visible.

B. *The effect of solution accuracy on CQS*. To test the sensitivity of CQS to the clustering solution, we simulated data with $r = 25$, and compared CQS of the true partition with that of

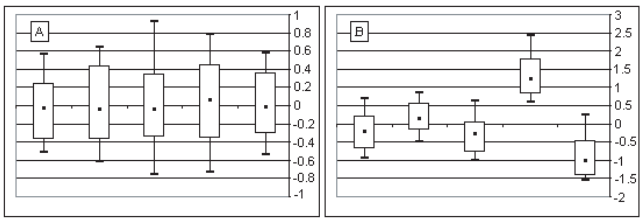


Fig. 2. Box plots for the projection of five simulated clusters with $r = 6$ (**A**) and $r = 25$ (**B**) after dimensionality reduction. The y-axis is the real-valued projection of the elements. Each box-plot depicts the median of the distribution (dot), 0.1 and 0.9 distribution quantiles (white box), and the maximum and minimum values.

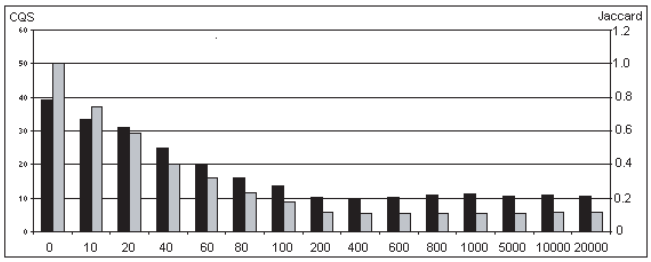


Fig. 3. Effect of the solution accuracy on CQS. The accuracy of different clustering solutions is measured by the number of inter-cluster exchanges introduced in the original solution. X-axis: number of exchanges. Y-axis: CQS (black bars, left scale) and Jaccard coefficient (gray bars, right scale).

other, similar and remote partitions. Those were produced by starting with the true solution and repeatedly exchanging a randomly chosen pair of elements from different clusters. As evident from the results in Figure 3, CQS is highest for the true partition and decreases with the number of exchanges applied (200 exchanges generate an essentially random partition, so further exchanges have no effect). We also computed for each intermediate solution its Jaccard score. As expected, the Jaccard coefficients of these solutions decrease with the number of exchanges.

C. *Sensitivity of CQS to the number of clusters*. Our next goal was to test the sensitivity of CQS with respect to the number of clusters. A robust score is essential for comparing solutions with different number of clusters. To this end we tested how CQS changes when splitting or merging clusters. For the splitting test we simulated data with $r = 25$. We compared the true 5-cluster solution with a 25-cluster solution obtained by randomly splitting each of the 5 clusters into 5 equal-size sub-clusters. This test was repeated 10 times. The parameters of the solutions before and after the splitting, averaged over 10 runs, are shown in Figure 4A. In all runs, as well as on the average, we observe a decrease of the clustering quality measures. The decrease of S/H is maintained (and even made more pronounced) in CQS and on the reduced data (S^*/H^*).

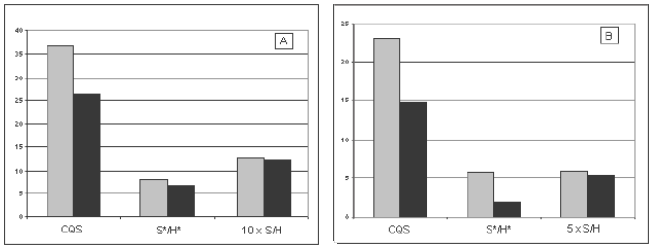


Fig. 4. Comparison between clustering solutions on simulated data after splitting clusters (**A**) or merging clusters (**B**). Each diagram shows CQS, S^*/H^* and S/H (y-axis) of the true solution (gray) and the modified solution (black). Numbers are average of 10 runs.

For the merging test, we simulated two 5-cluster data sets with $r = 25$ and $r = 6$ as above, using $n = 25$. We then combined these data sets into a single data set whose true solution consists of 10 equal size clusters with 25 genes each. We next merged pairs of clusters, one from each original data set, to form in total five clusters with 50 genes each. These five clusters comprised the alternative (merged) solution. Figure 4B shows the parameters of the resulting partition before and after the merging, averaged over 10 runs. As in the splitting test, all the measures decrease due to the merging, and this is observed in all runs, as well as on the average. The decrease of S/H is maintained and enlarged in S^*/H^* and CQS.

Next, we tested the agreement of CQS with Jaccard coefficient: we simulated 5-cluster data with $r = 25$ and applied K-means (MacQueen, 1965; Ball and Hall, 1967) to the data, with $K = 2, \ldots, 15$. Since K-means seeks a clustering solution with K clusters, we expect the solution's quality to decline as the difference $|K-5|$ increases. A good score should manifest such trend. We computed CQS and Jaccard coefficient for each clustering solution, as well as S/H. The results are shown in Figure 5. CQS behaves as the Jaccard coefficient and S/H, with a maximum at $K = 5$, the true number of clusters. Moreover, the ranking of all 14 solutions according to the Jaccard score (which is based on the true solution) and according to CQS (which is based on the attributes only) are virtually identical. The ratio score also does quite well, with a maximum at $K = 5$. However, the ranking of solutions by this score does not agree with the Jaccard score.

D. *CQS ability to detect fine clustering structures*. Our next goal was to test the ability of CQS to identify fine structures in the data. Profiles of 30 binary attributes were generated for four clusters of $n = 50$ genes each. For each attribute, its frequencies in clusters 1, 2, 3 and 4 were set to 2, b, $50-b$ and 48, respectively. We simulated data sets with $b = 3, 5, 10, 15, 20$. For each data set, we scored two clustering solutions: the original 4-cluster solution, and a 2-cluster solution obtained by merging cluster 1 with 2 and merging cluster 3 with 4. Thus, for large values of b we expect the 4-cluster solution to

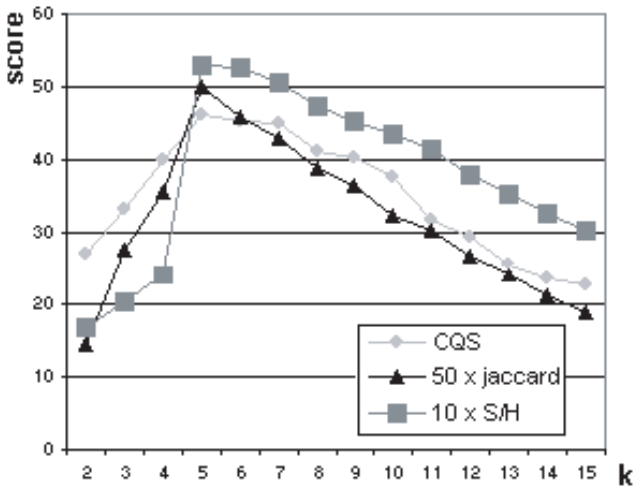


Fig. 5. Comparison of quality measures on solutions of various accuracies. Scores are plotted for different K-means' solutions. X-axis: K-means' solutions with $K = 2, \ldots, 15$. Y-axis: CQS (light gray), Jaccard (black) and S/H (gray) scores. The true number of clusters is 5.

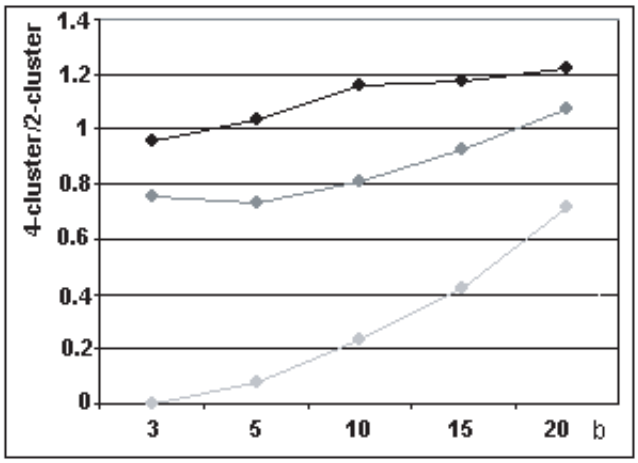


Fig. 6. Ability of the different scores to distinguish similar clusters. We simulated 4-cluster data with attribute frequencies $2, b, 50-b, 48$, and used different values for b. We obtained a 2-cluster solution by merging cluster 1 with 2 and 3 with 4. X axis: value of b in the simulation. Y-axis: the ratio of the scores for the 4-cluster and 2-cluster solutions. The scores are silhouette (gray), S/H (dark gray) and CQS (black).

score higher than the 2-cluster solution. Note that unlike the previous simulations, where distributions of individual attributes were designed to differ between clusters, here it is only the overall attribute density which is directly controlled. This design is the binary equivalent to the Gaussian clusters with different means that appears, e.g. in Pollard and van der Laan (2002). For each data set and each of the two solutions, we computed S/H, CQS and the average silhouette score.

The ratios of the 4-cluster to 2-cluster scores, averaged over 10 runs, are presented in Figure 6. As expected, the ratios are increasing with b in all scores. The silhouette for the 2-cluster solution is always greater than for the corresponding 4-cluster solution. Similarly, for $b = 3, 5, 10, 15$, S/H is greater for the 2-cluster solution. In all those cases, the scores would prefer the incorrect, 2-cluster solution. In contrast, CQS is able to identify the fine structure in the data: for all b values except $b = 3$, CQS rates the 4-cluster solution above the 2-cluster solution, as desired. For $b = 3$, the 2-cluster CQS is higher than the 4-cluster CQS, since there is almost no difference between the clusters with 2 or 3 occurrences of attributes, and between the clusters with 47 or 48 occurrences.

Yeast cell-cycle data

We also tested our approach on clustering solutions computed on the yeast cell-cycle data set of Spellman *et al.* (1998). The data set contains 72 expression profiles from yeast cultures synchronized by four independent methods: α factor arrest, arrest of a cdc15 temperature sensitive mutant, arrest of a cdc28 temperature sensitive mutant and elutriation. [As in Tamayo *et al.* (1999), an additional 90 min data point in the cdc15 experiment was not used.] Spellman *et al.* (1998) identified in these data 800 genes that are cell-cycle regulated. We used the expression levels of 698 out of those 800 genes, which have up to three missing entries, over the 72 conditions. The missing entries in each gene were completed with the average of its present entries. Each row of the 698×72 matrix was normalized to have mean 0 and variance 1.

Based on the analysis conducted by Spellman *et al.* (1998), we expect to find in the data five main clusters, each one corresponding to genes peaking in one of the cell cycle phases (G1, S, G2, M and M/G1). The 698×72 data set was clustered using four clustering methods: K-means (MacQueen, 1965; Ball and Hall, 1967), SOM (Kohonen, 1997; Tamayo *et al.*, 1999), CAST (Ben-Dor *et al.*, 1999) and CLICK (Sharan and Shamir, 2000; Sharan *et al.*, 2003). The solutions of K-means, SOM and CLICK were obtained using the EXPANDER software (Sharan *et al.*, 2003). CAST's solution was produced by the authors of the software and is the same as reported in (Shamir and Sharan, 2002). The K-means algorithm was executed with $K = 5$. The SOM algorithm was executed on a 2×3 grid and produced six clusters. The CAST solution has five clusters. CLICK was executed with default parameters and generated a solution with six clusters and 23 singletons. Each singleton was subsequently assigned to its closest cluster in order to produce a solution with no singletons. The similarity measure used in all cases was Pearson correlation coefficient. Another solution that we included in the analysis is the one reported in Spellman *et al.* (1998), which was generated by manually dividing the genes into five groups using their peak of expression, in order

to approximate the five cell-cycle phases. We shall refer to it as the 'true' solution.

To evaluate the five solutions, we used as gene attributes the GO classes (The Gene Ontology Consortium, 2000) at level 5 of the ontology, including process, function and component attributes. In addition, we used the MIPS annotation (Mewes *et al.*, 2002) at level 4. We removed attributes indicating that the functional class of the gene is still unknown and used only attributes that occur in at least four of the genes. Overall we used 51 GO process attributes, 37 GO function attributes, 27 GO component attributes and 59 MIPS attributes. We applied the analysis to 370 genes that had at least one attribute. CQS was computed three times, using the GO process attributes only, all GO attributes, and the MIPS attributes only. The results are depicted in Figure 7. For comparison purpose, we also scored a random clustering of the data into five equal-size clusters.

The random solution consistently obtained the lowest scores in all annotation categories. Using the process GO annotation (Fig. 7A), the CLICK, CAST and SOM solutions achieved the highest scores. Notably, they are scored higher than the 'true', K-mean and random solutions. When using all GO annotations (Fig. 7B), a similar pattern of scores is observed. Qualitatively, we got the same results when using GO annotations at level 4 of the hierarchy (data not shown). When evaluating all solutions based on MIPS level 4 annotations (Fig. 7C), CAST achieved the highest score. This exemplifies the fact that different biological attributes lead to different evaluations of clustering solutions.

In a different test, we ran SOM with $2, 3, \ldots, 8$ clusters on the same data set and calculated CQS of each solution. Clear best results were obtained for 5 and 6 clusters, as expected, (28 ± 1, 29 ± 1 respectively, with all other cluster solutions scored below 23).

Next, we present an analysis of CQS for the CLICK solution using all 115 GO attributes. Figure 8A is a scatter plot of weight versus enrichment for each attribute, using this solution. The *enrichment* of a k-cluster solution for a given attribute, is defined by $-\log p$ where p is the p-value of the G-test of independence (Sokal and Rohlf, 1995) with a $2 \times k$ table. It tests independence between element attributes and the partition into clusters. Note that the G-test enables us to evaluate functional enrichment for more than a single cluster. The frequently used hyper-geometric Fisher exact test of independence (Sokal and Rohlf, 1995) tests functional enrichment of a single cluster only.

As expected, the highest ranking attributes (both using the weights and the enrichment) are related to cell cycle. Notably, there is a correlation between the enrichment of an attribute and the absolute value of its assigned weight (Fig. 8B). This correlation is expected, since more enriched attributes can contribute more to our ability to discriminate between the clusters and, thus, they are expected to have higher weights. However, we do not expect a perfect correlation between the

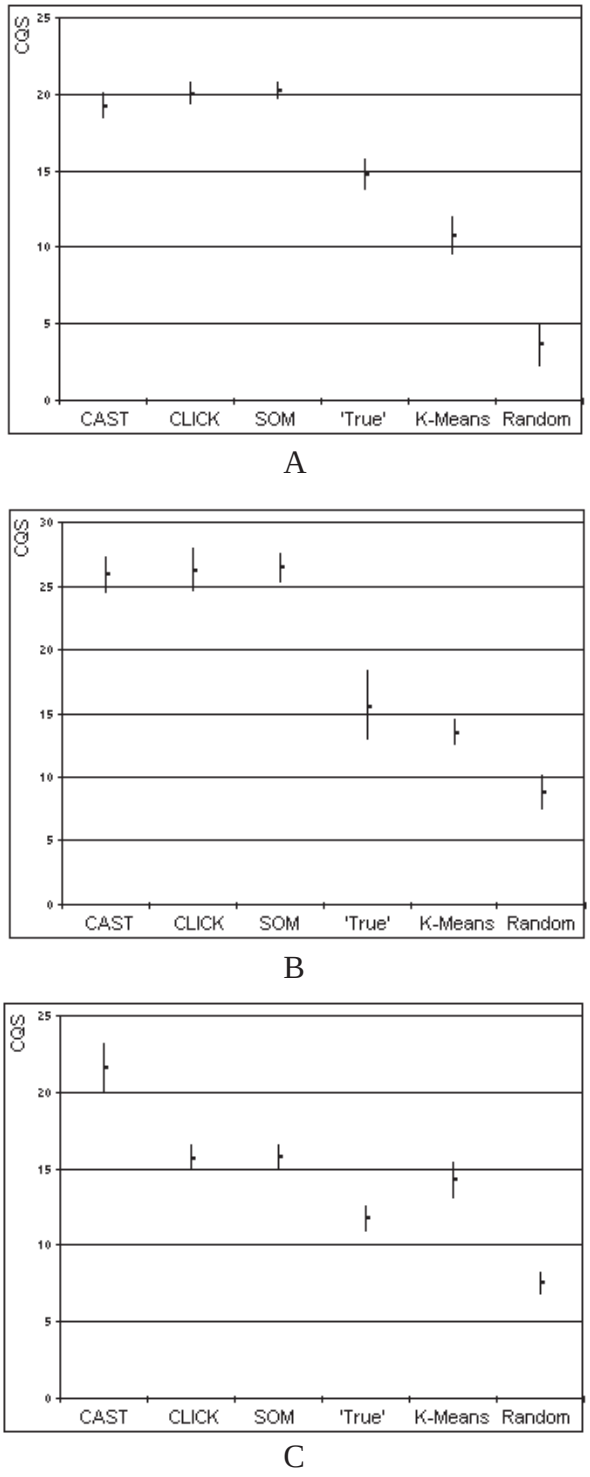
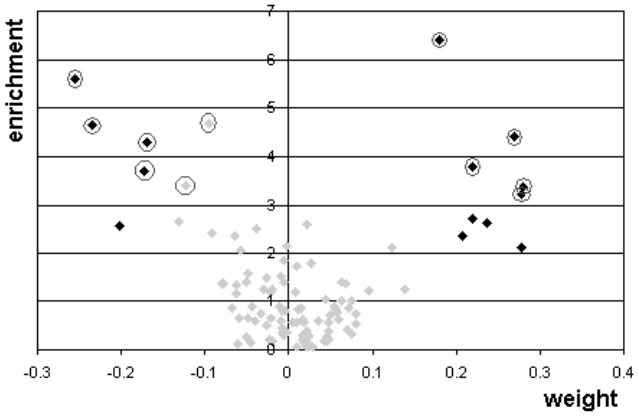


Fig. 7. CQS of six clustering solutions for the yeast cell cycle data of Spellman *et al.* (1998). CQS is computed using GO level 5 process attributes only (**A**), all GO level 5 attributes (**B**) and MIPS level 4 attributes (**C**). Y-axis: CQS. X-axis: clustering solutions. CQS for each clustering solution is presented along with its confidence, by computing the standard deviation of 10 other solutions achieved by seven random pair exchanges in the original solution.



A

weight	enrichment	Go Name	GO number
0.18	6.40	DNA metabolism	GO:0006259
-0.25	5.61	DNA replication	GO:0006260
-0.10	4.70	sensory perception	GO:0007600
-0.24	4.67	nucleoplasm	GO:0005654
0.27	4.41	amino acid metabolism	GO:0006520
-0.17	4.31	ATP dependent DNA helicase	GO:0004003
0.22	3.79	chromatin	GO:0005717
-0.17	3.70	chromatin binding	GO:0003682
-0.12	3.41	hexose transporter	GO:0015149
0.28	3.38	microtubule organizing center	GO:0005815
0.28	3.24	spindle	GO:0005819
0.22	2.69	DNA binding	GO:0003677
-0.13	2.63	cell wall	GO:0005618
0.24	2.62	chromosome organization	GO:0007001
0.02	2.58	nucleotidyltransferase	GO:0016779
-0.20	2.55	cytoplasm	GO:0005737
-0.04	2.51	transport	GO:0006810
-0.09	2.41	monosaccharide transport	GO:0015749
0.21	2.35	structural constituent of cytoskeleton	GO:0005200
-0.06	2.33	zygote formation (sensu Fungi)	GO:0030462
0.00	2.12	endoplasmic reticulum	GO:0005783
0.28	2.11	organelle organization and biogenesis	GO:0006996

B

Category/Cluster	1	2	3	4	5	6
DNA Metabolism (i)	23	11	2	1	1	0
DNA Replication (ii)	17	1	3	2	0	6
Chromosome organization (iii)	4	10	0	1	1	0
(i)+(ii)	10	0	1	1	0	0
(i)+(iii)	4	9	0	0	0	0
(ii)+(iii)	0	0	0	0	0	0
Total genes in clusters	101	76	45	94	32	22

C

Fig. 8. Attribute weights and enrichment values in the CLICK solution to the cell cycle data of Spellman *et al.* (1998), using all GO attributes. (**A**) A scatter plot of enrichment (y-axis) versus weight (x-axis), for each GO attribute. Attributes with high absolute weights (>0.15) are marked in black. Attributes with high enrichment (>3) are circled. (**B**) The 22 most enriched attributes. High attribute values in enrichment (>3) or weight (>0.15) are highlighted. Note that the 14 top weighted attributes are contained in the 22 most enriched attributes. (**C**) The distribution and co-occurrence of the attributes 'DNA metabolism', 'DNA replication' and 'Chromosome organization and biogenesis' in the six clusters of the CLICK solution.

two measures, since the goals of the attributes weighting and the enrichment measure are different and, more importantly, because the G-test takes into consideration each attribute separately, while the weights are computed by considering all attributes together and, thus, they reflect relations between attributes. For example, consider the 'DNA metabolism' attribute, which deviates significantly from the correlation (Fig. 8C). The enrichment of 'DNA metabolism' in clusters 1 and 2 overlaps to a large extent with that of 'DNA replication' and 'Chromosome organization and biogenesis', and this is partially reflected in their weights. Therefore, the weight of 'DNA metabolism' is lower than expected.

DISCUSSION

Clustering is a central tool in gene expression analysis. Different clustering methods usually produce different solutions, of which one has to pick one or few preferred solutions. We propose here a method called CQS for evaluating a clustering solution based on its biological relevance. Our method can be applied to compare the functional enrichment of many biological attributes simultaneously in different clustering solutions. In addition, it may be applied to optimize the parameters of a clustering algorithm (e.g. to determine the number of clusters). The method is based on using attributes of the clustered elements, which are available independently from the data used to generate the clusters.

We empirically validated CQS using a variety of simulations. Our scoring method was shown to outperform previous numeric methods for clustering evaluation, including the separation to homogeneity ratio and the average silhouette measure. We also applied CQS to compare between different clustering solutions of the cell cycle data set of Spellman *et al.* (1998) using binary attributes from the GO and MIPS annotation databases.

According to our results, CQS is sensitive to small modification of the clustering solution and to changes in the simulation setting. In order to evaluate the significance of the difference in CQS between clustering solutions, we use a CQS confidence measure. For example, the CAST, CLICK and SOM solutions in Figure 7A and B, cannot be meaningfully ranked by their scores. We may only conclude that CAST, CLICK and SOM have higher scores than the 'True', Random and K-means solutions. According to the results, although the 'True' solution was hand crafted in order to approximate the cell cycle phases, the solutions produced by CAST, CLICK and SOM are more aligned with the biological attributes. We note that these results should be treated with caution since the database annotations are incomplete and may be biased.

The attribute weights were computed using information about all the attributes together, without assuming that the attributes are independent. Frequently, the functional enrichment of each attribute in each cluster, is computed separately

[e.g. Tavazoie *et al.* (1999)]. In such cases, since the attributes might be dependent (as we exemplify in Fig. 8B), the real fraction of functionally enriched attributes might be over estimated.

CQS can be applied to a wide range of other attribute types. For example, one can use continuous attributes corresponding to sequence motifs, that represent the likelihood of having that motif. CQS has the advantage that it can use such continuous data without any assumption on the data distribution.

ACKNOWLEDGEMENTS

This study was supported in part by a research grant from the Ministry of Science and Technology, Israel. I.G.-V. was supported by the Colton Foundation. R. Sharan was supported by a Fulbright grant.

REFERENCES

Ball,G. and Hall,D. (1967) A clustering technique for summarizing multivariate data. *Behav. Sci.*, **12**, 153–155.

Ben-Dor,A., Shamir,R. and Yakhini,Z. (1999) Clustering gene expression patterns. *J. Comput. Biol.*, **6**, 281–297.

Ben-Hur,A., Elisseeff,A. and Guyon,I. (2002) A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.*, 6–17.

Bishop,Y., Fienberg,S. and Holland,P. (1975) *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, Cambridge, MA.

Dudoit,S. and Fridlyand,J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol.*, **3**, 0036.1–0036.21.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *PNAS*, **95**, 14863–14868.

Everitt,B. (1993) *Cluster Analysis*. 3rd edn. Edward Arnold, London.

Hansen,P. and Jaumard,B. (1997) Cluster analysis and mathematical programming. *Math. Program.*, **79**, 191–215.

Hartigan,J. (1975) *Clustering Algorithms*. Wiley, New York.

Huberty,C. (1994) *Applied Discriminant Analysis*. Wiley, New York.

Katz,B. and McSweeney,M. (1980) A multivariate Kruskal–Wallis test with post hoc procedures. *Multivariate Behavioral Res.*, **15**, 281–297.

Kaufman,L. and Rousseeuw,P. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley New York.

Kohonen,T. (1997) *Self-Organizing Maps*. Springer, Berlin.

MacQueen,J. (1965) Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. pp. 281–297.

McLachlan,G.J. (1987) On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Appl. Stat.*, **36**, 318–324.

McLachlan,G.J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York.

Mendez,M., Hoedar,C., Vulpe,C., Gonzales,M. and Cambiazo,V. (2002) Discriminant analysis to evaluate clustering of gene expression data. *FEBS Lett.*, **522**, 24–28.

Mewes,H.W., Frishman,D., Guldener,U., Mannhaupt,G., Mayer,K., Mokrejs,M., Morgenstern,B., Munsterkotter,M., Rudd,S. and Weil,B. (2002) MIPS: a database for genomes and protein sequences. *Nucleic Acid Res.*, **30**, 31–4.

Pesarin,F. (2001) *Multivariate Permutation Tests*. Wiley, New York.

Pollard,K. and van der Laan,M. (2002) A method to identify significant clusters in gene expression data. In *Sixth World Multiconference on Systemics, Cybernetics, and Informatics*, to appear.

Shamir,R. and Sharan,R. (2002) Algorithmic approaches to clustering gene expression data. In Jiang,T., Smith,T., Xu,Y. and Zhang,M. (eds.), *Current Topics in Computational Biology*. MIT Press, Cambridge, MA, pp. 269–299.

Sharan,R., Elkon,R. and Shamir,R. (2002) Cluster analysis and its applications to gene expression data. In Mewes,H.-W., Seidel,H. and Weiss,B. (eds.), *Bioinformatics and Genome Analysis*. Springer, Berlin, pp. 83–108.

Sharan,R., Maron-Katz,A. and Shamir,R. (2003) Click and expander: a system for clustering and visualizing gene expression data. *Bioinformatics*, in press.

Sharan,R. and Shamir,R. (2000) CLICK: a clustering algorithm with applications to gene expression analysis. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*. pp. 307–316.

Sokal,R. and Rohlf,F. (1995) *Biometry*. Freeman, San Fransisco.

Sokal,R.R. (1977) Clustering and classification: background and current directions. In Van Ryzin,J. (ed.), *Classification and Clustering*. Academic Press, London, pp. 1–15.

Spellman,P.T., Sherlock,G., Zhang,H.Q., Iyer,V.R., Andres,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Stephanopoulos,G., Hwang,D., Schmitt,W., Misra,J. and Stephanopoulos,G. (2002) Mapping physiological states from microarray expression measurments. *Bioinformatics*, **18**, 1054–1063.

Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tavazoie,S., Hughes,J.D., Campbell,M.J., Cho,R.J. and Church,G.M. (1999) Systematic determination of genetic network architecture. *Nat. Gene.*, **22**, 281–285.

The Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. *Nat. Gene.*, **25**, 25–29.

Tibshirani,R., Walther,G. and Hastie,T. (2000) Estimating the number of clusters in a dataset via the gap statistics. Technical report, Stanford University, Stanford.

Yeung,K., Haynor,D. and Ruzzo,W. (2001) Validating clustering for gene expression data. *Bioinformatics*, **17**, 309–318.

BIOINFORMATICS

Vol. 19 Suppl. 1 2003, pages i108–i117
DOI: 10.1093/bioinformatics/btg1014

Chain functions and scoring functions in genetic networks

I. Gat-Viks and R. Shamir*

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

One of the grand challenges of system biology is to reconstruct the network of regulatory control among genes and proteins. High throughput data, particularly from expression experiments, may gradually make this possible in the future. Here we address two key ingredients in any such 'reverse engineering' effort: The choice of a biologically relevant, yet restricted, set of potential regulation functions, and the appropriate score to evaluate candidate regulatory relations.

We propose a set of regulation functions which we call chain functions, and argue for their ubiquity in biological networks. We analyze their complexity and show that their number is exponentially smaller than all boolean functions of the same dimension. We define two new scores: one evaluating the fitness of a candidate set of regulators of a particular gene, and the other evaluating a candidate function. Both scores use established statistical methods. Finally, we test our methods on experimental gene expression data from the yeast galactose pathway. We show the utility of using chain functions and the improved inference using our scores in comparison to several extant scores. We demonstrate that the combined use of the two scores gives an extra advantage. We expect both chain functions and the new scores to be helpful in future attempts to infer regulatory networks.

Contact: {iritg,rshamir}@post.tau.ac.il

INTRODUCTION

The regulation of mRNA transcription is critical to cellular function. Large-scale gene expression (GE) measurements, using, e.g. DNA microarrays (Derisi *et al.*, 1997; Lockhart *et al.*, 1996), may enable the reconstruction of the regulatory relations among genes. By the *regulatory relation* of a target gene, we mean the set of genes that together regulate it, and the particular logical function by which this regulation is determined. This paper focuses on inference of regulatory relations from GE profiles.

Most current expression analysis tools are based on clustering (e.g. Eisen *et al.* (1998), Ideker *et al.* (2001) and Sharan *et al.* (2002)). Such analyses successfully reveal genes that are co-regulated, but not their regulatory relations. More advanced approaches rely on mathematical models of the regulation process. Different models at various levels of detail have been suggested. These include boolean (Ideker *et al.*, 2000; Akutsu *et al.*, 1999; Liang *et al.*, 1998), qualitative (Thieffry and Thomas, 1998), linear (Dhaeseleer *et al.*, 1999), differential equations (Chen *et al.*, 1999) and detailed biochemical models (Arkin *et al.*, 1998).

A key obstacle in the inference of regulation relations is the large number of possible solutions, and consequently the unrealistically large amount of data needed to identify the right one. This inherent complexity of genetic network inference (Akutsu *et al.*, 1998, 1999) led researchers to seek ways around this problem. Ideker *et al.* (2000) studied how to dynamically design experiments so as to maximize the amount of information extracted. Friedman *et al.* (2000) used Bayesian networks to reveal only parts of the genetic network which are strongly supported by the data. Hanisch *et al.* (2002) and Ideker *et al.* (2002) used prior knowledge about the metabolic network structure in order to identify relevant processes in GE data. Another approach to tackle the complexity issue is to reduce the set of allowed network models. Tanay and Shamir (2001) suggested a method of 'network expansion', in which one starts from a partially known network and augments it according to the GE data. Pe'er *et al.* (2002) make certain biologically-motivated assumptions on the local topology of the network, which reduce the space of possible global networks. Several other works used restrictive models of regulation relations (e.g. decision trees (Segal *et al.*, 2001)).

In this paper, we study two nuclear problems in regulation relation inference, which are at the heart of inferring transcription networks: (1) determining the set of regulators of a gene (the gene is called *regulatee* and the set is called its *regulators set*), and (2) deducing the precise mathematical function by which the regulators set determines the gene's transcription (the *regulation function*). We assume throughout a boolean model, i.e. each of the candidate regulators and regulatees can be in one of two

*To whom correspondence should be addressed.

states: expressed (present) or non-expressed (absent). The inference of regulatory relation of a single gene is a fundamental step in the long-term effort to infer regulation networks.

To study these problems we design two new methods which evaluate how well a candidate regulatory relation of a particular regulatee fits experimental data. Such *fitness scores* are essential in order to pick the right relation among many candidates. Our first score evaluates the specificity of the regulators set. The second score evaluates how well a particular regulation function (for a given regulators set) fits the data. Both scoring functions utilize established statistical methods, and are expressed as p-values, and thus are not very sensitive to over-fitting. Moreover, due to the Gaussian shape of these scoring functions, they always score only a few solutions at the high end. The two scores are affected differently by different problem parameters, so using both scores in combination gives an added advantage.

The second component of this work is the introduction and study of a novel family of regulation functions called *chain functions*. In a chain function, the state of the regulatee depends on the influence of its direct regulator, whose activity may in turn depend on the influence of another regulator, and so on in a chain of dependencies (we will provide formal definitions later). The class of chain functions has several important advantages: First, as we shall argue, these functions reflect common biological regulation behavior, and often occur in networks, so many real biological regulatory relations can be elucidated using them. Second, as we shall show, the number of chain functions with n control variables is $\Theta(n! \cdot (\log_2 e)^{n+1})$. This number is exponentially smaller than the total number of boolean functions. Hence, by limiting inference to chain functions, we reduce exponentially the size of the candidate solution search space.

We apply our approach to transcription profiles of the yeast galactose pathway (Ideker *et al.*, 2001). First, we demonstrate the advantages of using chain functions instead of searching through all boolean functions. Second, we use the yeast galactose pathway of Ideker *et al.* (2001) to compare our scores to several other fitness scores which were previously proposed for network inference, and show that on these data, our score outperforms them. Third, we show that by using in combination our two scores for regulator set and regulation function, we can obtain very high ranking of the correct solution.

The paper is organized as follows. We start by providing a formal framework for the model. We then define the chain functions, motivate them biologically and present their analysis. Next, the fitness scores are presented and analyzed. Finally, results on real transcription profiles are reported.

THE NETWORK MODEL

In this section we describe the formal model for our analysis and tools. The formalism follows Tanay and Shamir (2001) and Liang *et al.* (1998).

The set of all variables is denoted by U. These may include genes, mRNAs, proteins and ligands such as disaccharides and amino acids. The set of *states* that each variable in U may attain is denoted by V. A candidate regulation function for a variable g which is regulated by n variables $R_n \subseteq U$, has the form $f^g : V^n \rightarrow V$. In other words, the state of g is a function of the states of the variables in R_n. We use the term *regulatee* for the regulated variable g, and the term *regulator* (of g) for each variable in R_n. The regulator set may actually include biological regulator, co-regulators, co-factors, etc.

The GE data consist of l conditions, $E = \{e_1, \ldots, e_l\}$. Condition j is defined by a vector of levels (typically expression ratios) for each variable in U, and by a set of variables that were externally perturbed (knocked-out or over-expressed) in condition j. These externally perturbed variables must be indicated, as their levels are not determined by their regulation functions. We assume that the data are of steady state, so additional synchrony assumption is not needed, and the states of the regulators determine the state of the regulatee in the *same* condition. A simple modification of the model applies to time-series synchronous data, where the state of the regulatee is taken at one time point later than that of the regulators (cf. Tanay and Shamir (2001)).

We will narrow the range of network models by adding constraints as follows: We assume that the states are discrete, and that the functional relations are deterministic. Each variable can have only two levels: either on (1) or off (0), i.e. $V := \{0, 1\}$. This can be achieved, for example, by setting a threshold on the input data values. We shall use $state(x, j)$ to denote the binary value of variable x in condition j, and suppress j whenever possible for readability. Each regulatee is regulated through a boolean function of at most n arguments. The boolean model is a drastic simplification of real biology, yet it captures important features of biological systems. Similar simplifying choices are frequently made in order to reduce the number of degrees of freedom, and to avoid over-fitting (cf. Akutsu *et al.* (1999) and Kauffman (1974)).

CHAIN FUNCTIONS

We now propose a class of regulation functions, called chain functions. We argue that this class covers many common regulation scenarios in biology. We analyze the chain functions and show that the set of chain functions is exponentially smaller than the set of all boolean regulation functions.

Definitions. We first define some related terms. Recall that the state of variable in a condition is 1 if that variable is present and 0 if it is absent. The chain function f^{g_0} on the variables $g_n, \ldots, g_1$ will determine the value of the regulatee g_0. The order of the variables is important, as it reflects the order of influence among them, as will be explained below. For that reason, we shall sometimes refer to R_n as the ordered set $g_n, \ldots, g_1$. We call g_i the *predecessor* of g_{i-1} and the *successor* of g_{i+1}. f^{g_0} depends on n auxiliary *control bits* $c_n, \ldots, c_1$ that attain values A or R. The semantic is that $c_i = A$ (R) if g_i activates (represses) g_{i-1}. These two options are exhaustive. Note that the activation or repression by g_i is of g_{i-1} and not of the regulatee g_0. We also call $c_i = A$ and $c_i = R$ *positive* and *negative control*, respectively.

The control bit c_i defines whether a regulator g_i is a repressor or an activator of its successor g_{i-1}. However, this effect takes place only if the regulator g_i is currently active. Consider, for example, a regulator g_2 with control bit A. g_2 will activate g_1, but only if g_2 is actually active. Inactivity may be due to its absence, or g_2 might be present and inactive, if it is repressed by its predecessor g_3. To define this situation, we use two concepts: the *activity* of a variable $a(g_i)$ and its *influence* on its successor $infl(g_i)$. Activity can be either 0 or 1; influence can be either positive (P) or negative (N). Their definitions are recursive. The influence on g_n is always positive. Formally, $infl(g_{n+1}) = P$. The activity of g_i is 1 iff the influence on it is positive and its state is 1:

$$a(g_i) = 1 \; iff \; (infl(g_{i+1}) = P \; and \; state(g_i) = 1) \tag{1}$$

The influence of g_i on g_{i-1} is defined by:

$$infl(g_i) = P \; iff \; \begin{cases} c_i = A \;\; and \;\; a(g_i) = 1, \; or \\ c_i = R \;\; and \;\; a(g_i) = 0 \end{cases} \tag{2}$$

Equivalently, $infl(g_i) = N$ iff $[c_i = A$ XOR $a(g_i) = 1]$. Finally, the state of the regulatee g_0 is simply the influence of g_1: $f^{g_0}(g_n, \ldots, g_1) = 1$ iff $infl(g_1) = P$.

Even if g_0 is regulated by the function f^{g_0}, usually, due to experimental noise, not all conditions will manifest f^{g_0}. We say that condition j is *consistent* with f^{g_0} if $state(g_0, j) = f^{g_0}(g_n, \ldots, g_1)$, where the states of $g_n, \ldots, g_1$ are taken in condition j.

The *control pattern* of f^{g_0} is the binary vector $c_n, \ldots, c_1$. For example, RAARR is the control pattern for a function with $c_5 = c_2 = c_1 = R$ and $c_4 = c_3 = A$. The *state pattern* of the variables of f^{g_0} is $state(g_n), \ldots, state(g_1)$. For example, 10100 corresponds to $state(g_5) = 1, state(g_4) = 0$ etc.

Biological motivation. We present below several biological examples that explain the motivation for defining chain functions. The *Trp operon* of *E. Coli* is a classic example (Neidhardt, 1996). If the promoter of the *Trp* operon is bound by a repressor (TrpR), the expression of the tryptophan-producing enzymes is prevented. The blocking of expression is regulated in the following way: to bind to its promoter DNA, TrpR must have two tryptophan molecules (L-Trp) bound to it. This is an example of negative control, where removal of the ligand switches the *Trp* operon on. This example corresponds to a chain function with $n = 2$ (see Figure 1A), where g_0, the regulatee, is the *Trp* operon, g_1 is TrpR, and g_2 is L-Trp. c_2, the control bit of the L-Trp, is A, since L-Trp activates TrpR. $c_1 = R$, since TrpR represses the transcription of the regulatee. The activity of L-Trp (g_2) depends only on its presence. Thus, if L-Trp and TrpR are present (the state pattern is 11), then $a(g_2) = 1$ and thus $infl(g_2) = P$, which implies that $a(g_1) = 1$, and so $infl(g_1) = N$, so we expect no expression of g_0. One can compute similarly the expression level for any other state pattern.

Another well known example of a generic regulation switch is galactose utilization in the yeast *S. cerevisiae* (Jones *et al.*, 1992). This process occurs in a biochemical pathway that converts galactose into glucose-6-phosphate. The transporter gene gal2 encodes a permease that transports galactose into the cell. A group of enzymatic genes, gal1, gal7, gal10, gal5 and gal6, encode the proteins responsible for galactose conversion. The regulators gal4p, gal3p and gal80p control the transporter, the enzymes, and to some extent each other (Xp denotes the protein product of gene X). In the following, we describe the regulatory mechanism, assuming that glucose is absent in the medium. gal4p is a DNA binding factor that activates transcription. In the absence of galactose, gal80p binds gal4p and inhibits its activity. In the presence of galactose in the cell, gal80p binds gal3p. This association releases gal4p, so that gal4p actually activates transcription. This mechanism can be viewed as a chain function, where $(g_4, g_3, g_2, g_1) = (galactose, gal3, gal80, gal4)$, and the corresponding control pattern is *ARRA*. The known regulatees are gal1, gal7,gal10, gal5, gal6 and gal2 (see Fig. 1B).

In general, two fundamental mechanisms by which gene regulatory proteins control gene transcription are negative regulation via transcriptional repressors, and positive regulation via transcriptional activators. Inducing ligands can turn a gene 'on' by either activating transcriptional activator or repressing transcriptional repressor. Likewise, inhibitory ligands can turn 'off' a gene either by inactivating an activator or activating a repressor. These mechanisms are simple cases of chain functions. Examples in *Escherichia coli* include the *lac operon* repression by the λ repressor and lactose, *araBAD operon* activation by araC and arabinose, and the CAP activator in the presence of cAMP (Neidhardt, 1996). More complex regulation functions, such as the signal transduction controlling the SOS

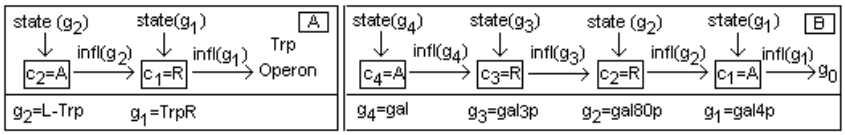


Fig. 1. Chain functions. (A) *Trp operon* regulation. (B) galactose pathway regulation.

response in *E.coli* (Neidhardt, 1996), and genes expression during the development of the drosophila's embryo (Mannervik *et al.*, 1999), might be also viewed as chain functions.

In more complex situations, one simple chain function may not be enough. Some systems should be modeled by several chains combined by boolean operators. (e.g. the general amino acids control chain, which operates in conjugation with the arginine specific regulatory chain (Jones *et al.*, 1992)). Several regulators which have the same functionality may be modeled as alternative regulators in a single node along the chain. (e.g. Fus3 and Kss1 in the *S. cerevisiae* pheromone response). In addition, we might need more levels of discretization. The key concept in chain functions is that activity level of a regulatee is determined by a chain of influences. This concept is not limited to a boolean model (see also concluding remarks). The chain functions as defined here can be used as basic building blocks for modeling more sophisticated regulation systems.

Direct effectors. Genetic networks are frequently represented as wiring diagrams, which show 'who regulates whom but not how'. The *direct effectors* of g_0 are defined as the minimal set of variables with the property that given any combination of their states, the state of g_0 is independent of any other variable. A *wiring diagram* is a directed graph in which the parents of a regulatee are its direct effectors. It is easy to see that every regulator in a chain function is a direct effector of the regulatee (a proof appears in Appendix A), and no variable outside $g_n, \ldots, g_1$ is a direct effector. An arrow in a chain function diagram reflects influence between regulators which are both direct effectors of g_0, and should not be confused with arcs in the wiring diagram, which represent direct transcription effect of the parent on the child.

Note that direct effectors are not necessarily limited to cis-regulatory elements (e.g. transcription factors and ligands) acting directly on the promoter of the regulatee. In fact, additional molecules with no direct connection or physical proximity to the promoter may be direct effectors, as demonstrated in the galactose example. Chain functions exemplify that very remote effectors can sometimes be included in the (so called) direct effectors set.

Chain layers. Any control pattern may be separated into *layers*, by truncating the control pattern after each R. For example, the pattern $ARRARAAAA$ has four layers: $l_4 = AR$, $l_3 = R$, $l_2 = AR$ and $l_1 = AAAA$. The first layer has two possible *layer types* $A \ldots A$ or $A \ldots AR$, and all other layers must have the type $A \ldots AR$. For brevity, the former will be called *type A* and the latter *type R*. Note that the number of As in a type R layer may be zero. Define a *permutation* on a chain function as a reordering of the regulators without changing the control pattern. For example, there are two different permutations for the chain function f with $R_2 = \{x, y\}$ and control pattern RR: (x, y) and (y, x). These two permutations yield different functions: If the states of x and y are 0 and 1 respectively, then $f(x, y) = 0$ and $f(y, x) = 1$. Similarly, if the control pattern is RA, the two permutations yield different functions. However, it is easy to verify that if the control pattern of f is AA or AR, the two permutations yield the *same* function. Thus, if x and y belong to the same layer, they can be permuted without changing the function, and otherwise their permutation yield different functions. This can be generalized as follows: Given a chain function $f^{g_0}(g_n, \ldots, g_1)$, define a *class* as a consecutive group of regulators out of $g_n, \ldots g_1$ that can be arbitrarily permuted while keeping the control pattern, without changing the function. We can show that the layers partition the regulators into a minimal number of classes (see Appendix A). This implies that the order of regulators inside layers is insignificant. Hence, we may focus on the interaction between layers. The incoming influence from the previous layer, and the states of regulators inside the layer, (in fact, the conjunction of their states), determine the outgoing influence of a layer on the next one.

Layers can be interpreted biologically as follows: In case the influence on the downstream elements depends on the cooperation of several factors, this part in the ordered chain constitutes a layer. Prominent examples are transcription factor complexes (e.g. Jones *et al.* (1992)) and the signal activation cascades (e.g. the MAPK cascade in yeast (Roberts *et al.*, 2000)). As another example, many arginine biosynthetic genes are regulated by arginine specific repression of $arg80$, $arg81$ and $arg82$, which constitute a type R layer (Jones *et al.*, 1992).

The number of chain functions. A trivial upper bound on the number of chain functions of n variables is $O(2^n \cdot n!)$. This follows since each control bit can be A or R, and there are $n!$ possible permutations of the variables. This bound is exponentially smaller than the total number of n-variable boolean functions, which is $\Theta(2^{2^n})$, but it ignores the equivalence classes formed by the layers. In Appendix A, we study the problem of counting the exact number of chain functions of n variables, and provide the following tight asymptotic bound:

THEOREM 1. *The number of chain functions with n control variables is* $\Theta(n! \cdot (\log_2 e)^{n+1})$.

For example, the total number of boolean functions is 256, 16500, $4.29 \cdot 10^9$ and $1.84 \cdot 10^{19}$ for $n = 3, 4, 5$ and 6, respectively. In contrast, the corresponding numbers of chain functions are 26, 150, 1082 and 9366. Thus, the set of chain functions is dramatically smaller than the set of all possible regulation functions. This allows more accurate inference of a function from expression data, if it is assumed to be a chain function.

SCORING FUNCTIONS

Assume the regulatee g_0 is fixed. Our goal is to find the best explanation for the regulation of g_0, given the expression data. This requires a score, or a *scoring function*, which evaluates how well a regulation function fits the data. Several scores, including mutual information, rSpec and BDE (see Results for more details) were suggested in previous studies. Here we propose and analyze two new scores: One evaluates a particular set of regulators of g_0, without attempting to determine the regulation function itself. The other evaluates a particular function for a given set of regulators. The scores are designed to test any regulators set or any candidate regulation function. In particular, the development and use of the scores are completely independent from our study of chain functions.

Regulators specificity. We first wish to evaluate the specificity of a set of regulators R_n to a certain regulatee g_0. We present here a hypothesis-testing approach to this question.

Let M be a matrix summarizing the expression data, where rows correspond to the $r = |V|$ states of g_0, and columns corresponds to the $c \leq r^n$ state patterns of R_n which appear in the data. m_{ij} is the number of co-occurrences of the ith state of g_0 with the jth state pattern of R_n in the same condition.

Consider the null hypothesis H_0 that the state of g_0 and the regulators' state pattern are independent. Rejection of H_0 indicates that the state of g_0 depends on the regulators' state pattern, so there is high correlation of the regulators and the regulatee. To test the hypothesis, we use the G-test of independence (Sokal and Rohlf, 1995). The logarithm of the *generalized likelihood ratio* statistic $\lambda(M)$ of the above hypothesis is $\ln\lambda(M) = -\sum_{i=1}^{r}\sum_{j=1}^{c} m_{ij} \cdot \log\frac{m_{ij}}{m} + \sum_{j=1}^{c} m_{\cdot j} \cdot \log\frac{m_{\cdot j}}{m} + \sum_{i=1}^{r} m_{i\cdot} \cdot \log\frac{m_{i\cdot}}{m}$, where $m_{\cdot j} = \sum_{i=1}^{r} m_{ij}$, $m_{i\cdot} = \sum_{j=1}^{c} m_{ij}$ and $m = \sum_{i=1}^{r}\sum_{j=1}^{c} m_{ij}$. A fundamental property of likelihood ratio tests in general is that the asymptotic null distribution of $-2\ln\lambda$ is $\chi^2_{t-t'}$, where the parameter space of $H_0 \cup H_1$ is t-dimensional and the parameter space of H_0 is t'-dimensional. This property is known as the Wilks phenomenon (Wilks, 1938). Accordingly, in our case the asymptotic null distribution of $-2\ln\lambda(M)$ is a nearly $\chi^2_{(c-1)\cdot(r-1)}$-distribution. Therefore, we define *regSpec*, the specificity of the set of regulators R_n for g_0, as the p-value that corresponds to the test statistic $-2\ln\lambda(M)$, and evaluate it using $\chi^2_{(c-1)\cdot(r-1)}$.

$\ln\lambda(M)$ is proportional to the mutual information between the regulators' state pattern and the regulatee: $\frac{-\ln\lambda(M)}{m}$ is precisely $I(x : y) = H(x, y) - H(x) - H(y)$ (cf. Cover and Thomas (1991)). Mutual information has been used in several studies of genetic networks (e.g. Liang *et al.* (1998), Pe'er *et al.* (2002) and Friedman *et al.* (2000)). regSpec has the advantage of assigning a probability to the mutual information expression.

regSpec measures the unevenness of the frequencies m_{ij} for each i. For a fixed number k of conditions, regSpec evaluates the way k is distributed among the $c \times r$ cells in M. When c is large, most cells will contain low frequencies and the unevenness will be low. Hence, regSpec has a bias towards small c values.

The size of matrix M defined above is bounded by 2^{n+1} for $r = 2$, so by a naive implementation of regSpec, the total cost of the computation for a given set of n regulators and the regulatee is $O(l \cdot n + 2^{n+1})$. The first part is the cost of building M and the second, of computing $\ln\lambda(M)$ and the χ^2 approximation. Since typically $n < 20$, this time is moderate in practice.

The fitness of a regulation function. We now wish to evaluate how well a particular regulation function fits the experimental data. Let S be a state pattern of the regulators R_n and let f^{g_0} be any regulation function. f^{g_0} determines the expected state of g_0 for the state pattern S. Given a set of conditions $E = \{e_1, \ldots e_l\}$, the *difference* vector Δ of a particular combination g_0, f^{g_0}, E, R_n is: $\Delta(S) = |\{e_j | state(R_n) = S, f^{g_0}(S) = state(g_0, j)\}| - |\{e_j | state(R_n) = S, f^{g_0}(S) \neq state(g_0, j)\}|$. Hence, Δ counts the number of agreements (consistent cases) minus the number of disagreements in the data with f^{g_0} for the pattern S. We shall refer to Δ of a particular combination g_0, f^{g_0}, E, R_n without explicitly specifying it. The size of the Δ vector is c, the number of different state patterns S.

Denote by d_0 the number of patterns S in the data with

$\Delta(S) = 0$. If e other $\Delta(S)$ values appear, let $d_1, \ldots d_e$ be the number of times each of them appear. Now, rank the absolute values of the difference vector and to the rank of each absolute value attach the sign of the difference in Δ. In case of a tie, rank by midranks, i.e., tied values are ranked by their mean rank. Let us denote the ranks whose signs are negative by $R_1 < \cdots < R_a$ and those with positive signs by $S_1 < \cdots < S_k$ so that $c = a + k + d_0$.

Consider now testing the hypothesis H_0 of no difference between the agreement and disagreement frequencies, against the alternative that there are more agreements then disagreements. Thus, rejection of H_0 is more likely if k is large and if the positive signed ranks tend to be larger than the negative signed ranks. The *Wilcoxon signed rank test* (Lehmann and D'abrera, 1975) offers a simple statistic that combines these criteria in the sum of the positive signed ranks $V_s = S_1 + \cdots + S_k$. H_0 is rejected where V_s is sufficiently large. We define $funcFit$ as the p-value that corresponds to the test statistic V_s. The p-value for V_s is available in the Wilcoxon standard signed rank table for the null distribution of V_s. Beyond the range of the table, one can use the normal approximation, where the expectation and the variance of V_s are $E_{H_0}(V_s) = \frac{c(c+1)-d_0(d_0+1)}{4}$ and $Var_{H_0}(V_s) = \frac{c(c+1)(2c+1)-d_0(d_0+1)(2d_0+1)}{24} - \frac{\sum_{i=1}^{e} d_i(d_i+1)(d_i-1)}{48}$. Note that funcFit uses the ranking of the differences only, and not their actual values. This makes it less sensitive to inconsistencies or noise.

For a given set of regulators, all possible regulation functions have the same absolute difference vector. Thus, for each set of regulators, we may compute once the absolute differences vector in $O(l \cdot |U|)$ and the midranks, expectation and variance in $O(c \log c)$ ($c \leq \min(2^n, l)$).

When searching the maximum V_s over all boolean functions, a single computation summing over all ranks of the non-zeros differences gives the answer in $O(c)$. However, when the set of functions is restricted, (e.g. when only chain functions are considered), V_s should be computed for each regulation function separately, since each regulation function characterizes a distinct sequence of ranks $(S_1, \ldots, S_k)$. There are $\Theta(2 \cdot n! \cdot (\log_2 e)^{n+1})$ chain functions, and we need $O(c)$ work in order to sum over $(S_1, \ldots, S_k)$ for each one, so the total cost for computing funcFit for chain functions is $O(c \cdot n! \cdot (\log_2 e)^{n+1})$.

The scoring scheme. When we wish to find the best regulatory relation, we can, in principle, find the best regulation set using regSpec, and then use funcFit to find the best function for that set. However, as discussed above, the two scores have different biases to errors, the amount of unevenness in each column of M, and c value. Hence, using the two scores together and seeking regulatory relations that score high in both is advisable.

RESULTS

To test our methods, we applied them to the yeast galactose pathway dataset of Ideker *et al.* (2001). Since high throughput data of protein levels are currently unavailable, we use the mRNA expression levels to model both transcription levels and the abundance of the proteins, assuming that the amount of mRNA presented in the cell is indicative of its protein levels. The dataset contains 23 expression profiles, each corresponding to some perturbation in the galactose pathway. Guided by the current galactose system model, wild-type and nine genetically altered yeast strains were examined, each with a complete deletion of one of the nine galactose pathway genes: gal2Δ, gal1Δ, gal5Δ, gal7Δ, gal10Δ, gal3Δ, gal4Δ, gal6Δ and gal80Δ. Each of the nine strains was also perturbed environmentally by growth in the presence of galactose (+gal), and in the absence of galactose (-gal). Additionally, three double perturbations were performed: gal80Δgal2Δ -gal, gal80Δgal4Δ -gal and gal10Δgal1Δ +gal. The reference to all these conditions is the wild-type, grown in +gal media. Ideker *et al.* computed for each gene and condition the mRNA expression ratio relative to the reference, and assigned to it a confidence value. We transformed the data into binary states as follows: For each gene and condition, if the confidence value was high (above 45 (Ideker *et al.*, 2001)), and the ratio was above 1 (below -1), we set the state value to 1(0). For low confidence values, we assumed the expression level was identical to the wild-type expression, and set the state to 1, since in the reference condition all the galactose system genes are expected to be expressed (in the presence of galactose and absence of glucose (Jones *et al.*, 1992)).

We used as the set of potential regulators gal4, gal3, gal80, gal1, gal2, gal5, gal6, gal7, gal10, gcn1 and galactose. As regulatees, we checked the genes gal1, gal7, gal10, gal2, gal5 and gal6, since their regulation has been well characterized previously (see Figure 1B). For analyzing a regulatee, we do not use data from strains with its complete deletion. We used $n = 4$ throughout.

We compared the performance of funcFit to the following alternative scores: (a) rSpec (Tanay and Shamir, 2001), which is essentially minus the logarithm of the p-value of the number of conditions for which the regulation function is consistent with the observed expression of the regulatee. (b) Mutual information (Cover and Thomas, 1991) between the observed expression level of the regulatee and the expected expression level generated by applying the regulation function on R_n. (Note that it scores a particular regulation function, unlike the mutual information mentioned in the Scoring Functions Section). (c) BDE with the following informative priors: $N'_{ijk} = \frac{N' \cdot 0.9}{2^n}$ for consistencies and $N'_{ijk} = \frac{N' \cdot 0.1}{2^n}$ for inconsistencies, where $N' = 10$, and with non-informative

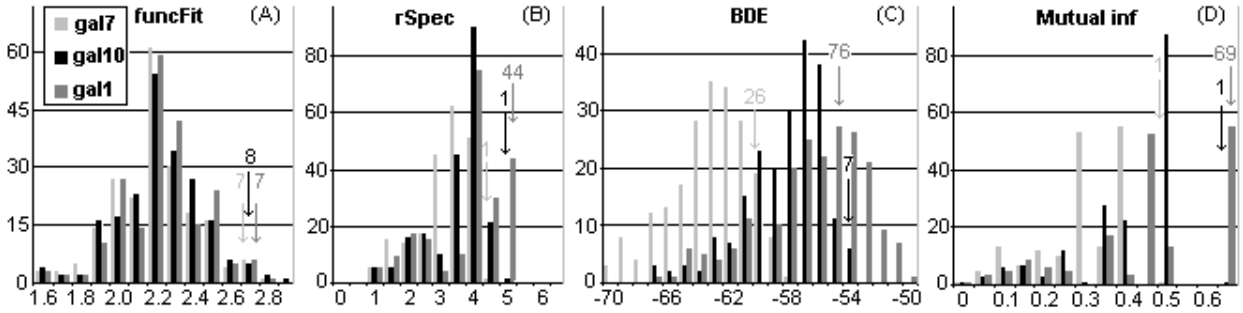


Fig. 2. Comparison of scoring methods. x-axis: The scores of funcFit (A), rSpec (B), BDE (C) and mutual information (D). y-axis: The number of regulators subsets R_4 whose best scored chain function attained that score. The regulatees are: gal7 (gray), gal10 (black) and gal1 (dark gray). The arrows mark the true regulation function solution for each regulatee, according to its color. The number above each arrow is the number of regulator sets whose scores are at least as high as the score of the real regulation function.

priors where $N^{'} = 1$. See (Heckerman *et al.*, 1995) for definitions and a description of BDE.

Our test was as follows: For each regulatee, we checked each possible subset of $n = 4$ regulators, and for each such subset, we checked every possible regulation function, and found the best scoring one. We performed this test twice, once using all boolean functions, and once using only chain functions. We repeated this test with the scoring functions BDE, mutual information, rSpec and funcFit. In all tests, we did not allow auto-regulation, and each regulator was allowed to appear only once along the chain function. Testing was done using a C++ software implementation written in-house. It can analyze all chain (boolean) functions with $|U| = 11$ and $n = 4$, for a single regulatee, in 30(15) seconds on a standard 800MHz PC.

In Figure 2, we present the performance of the different scoring methods. For each candidate regulation set, all chain functions are scored and the best score is presented. As can be seen in the figure, mutual information and rSpec tend to score high a large portion of the regulator sets. Thus, occasionally they may infer a lot of false positive regulation functions. Moreover, a small difference in the consistency level can cause a regulation function to be ranked very high or very low. In mutual information, there are 1, 1 and 69 regulator sets whose best chain functions are in the highest score category, for gal1, gal7 and gal10 respectively. In rSpec, there are 1, 1 and 44 chain functions in the highest scores category, for the same regulatees. The main reason for this instability is that both scores take into consideration only the total number of consistent conditions, without considering their state patterns at all. Unlike these scores, BDE and funcFit consider the distribution of inconsistency among the state patterns. Moreover, funcFit and BDE have a Gaussian-like distribution of scores, which is preferable, since we always get only a few top scoring candidates. Nevertheless, BDE does not always rank the real chain function high: In BDE, there are 26, 76 and 7 chain functions above the real solution, for the three regulatees. Since BDE penalizes state patterns with noise, but does not penalize for missing state patterns, it has a bias to small c values. This may explain its poorer performance in comparison with funcFit. funcFit is the only score which consistently ranks the real solution high: there are only 7, 8 and 7 chain functions equal to or above the real solution, for the same three regulatees. Qualitatively similar results were obtained when repeating the same analysis with all boolean functions, and with the BDE score using different N' values as well as non-informative priors.

Our next test aimed to see the effect of restricting inference to chain functions only. In Figure 3 we present the maximum funcFit scores distribution for the chain functions set and for all boolean functions. As expected, when using all boolean functions, the distribution tends to spread to higher values. Moreover, by using chain functions only, the real solution is ranked higher: In gal10, there are 16 boolean functions and only 8 chain functions with scores equal to or above the real one. In gal1 and gal7, the corresponding numbers are 16 and 7. Qualitatively similar results were obtained using the other scoring methods. In principle, when using all boolean functions, the distribution may tend to spread much more drastically to high values. However, the specific dataset that we analyzed was not large enough to manifest this difference: Although there are theoretically 65,536 boolean functions and 150 chain functions, actually only 200 boolean functions and 40 chain functions are effectively different (on average), because on average only 7.5 different state patterns appear in the data (out of 16 possible ones) for each group of regulators. In larger datasets with more state patterns, the advantage of the chain functions should be more pronounced.

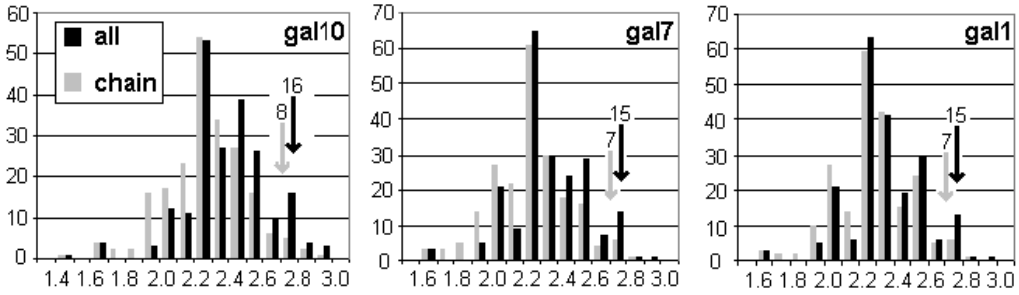


Fig. 3. Comparison of funcFit distribution for all functions and for chain functions only. x-axis: The funcFit score. y-axis: The number of regulators subsets with $n = 4$ that attained that maximum funcFit value. Maximum was computed and plotted among all boolean functions (black) and among chain functions only (gray). Results are reported for the regulatees gal10, gal7 and gal1 (from left to right). Arrows and numbers are as in Figure 1.

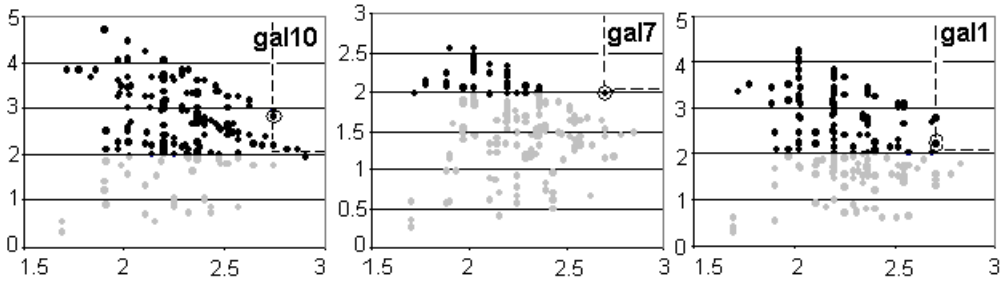


Fig. 4. A scatter plot of regSpec (y-axis) vs. funcFit (x-axis) scores, for each subset of regulators R_4. Relevant subsets of regulators ($regSpec > 2$) are in darker shade. The true solution is circled. The regulatees are (from left to right) gal10, gal7 and gal1. The broken lines indicate the quadrant above $regSpec = 2$ and funcFit equal to the true value.

Our next goal was to examine the advantage of using regSpec and funcFit together. The test used the same setup described above. Figure 4 is a scatter plot of the highest funcFit score (for chain functions) versus the corresponding regSpec score, for each subset of regulators. As expected, some subsets get a very high regSpec score and a low funcFit score, or vice versa. Also, there is some tradeoff between high specificity and high funcFit, probably due to their opposite preferences (see the Scoring Functions Section). Many of the candidate regulator subsets have a very low regSpec score, and thus we can reduce significantly the computing time by foregoing their funcFit computations altogether. By searching only above $regSpec = 2$, the true chain functions are ranked very high in funcFit. For regulatees gal10, gal7 and gal1, there are only 4, 0 and 1 alternative chain functions whose funcFit scores are the same or higher. In tests on the regulatees gal5, gal6 and gal2, all possible subsets of regulators have $regSpec < 2$, and thus we could not analyze their regulation functions. We suspect that these low regSpec values are due to the stringent discretization thresholds that we used.

CONCLUDING REMARKS

In this paper, we propose a biologically relevant class of regulation functions. We also suggest two scoring methods by which one can evaluate candidate regulatory relations, and demonstrate their advantage over extant scores. We tested our method on experimental gene expression data, in trying to infer gene regulation relations. We showed the utility of using chain functions, and the advantage of our scores over several extant methods.

Clearly, more extensive testing of our methods on additional datasets and pathways is needed. By tests on large datasets we expect to demonstrate the fuller advantage of using a restricted set of relevant regulation functions. We expect to identify more regulation functions and refine our results by allowing more than two levels of discretization and assigning a probability distribution over those levels. In addition, we expect that the special structure of chain functions can be exploited in the design of follow-up experiments.

The ability to score and restrict regulatory relations are fundamental components in the grand challenge of reconstructing regulatory networks. In order to extend this

work towards global network reconstruction, the chain function model should be extended. It should allow several chain functions combined by a boolean operator. Handling functions with unknown number of regulators should be addressed. Cases where there are several regulatees whose regulation chains have common parts, should also be considered.

ACKNOWLEDGEMENT

We thank Noga Alon for pointing us to the literature on Ordered Bell numbers. We thank Amos Tanay and Ori Gurel for helpful discussions. This study was supported by a pilot grant from the McDonnell foundation and by the Israeli Science Foundation (grant no. 309/02).

REFERENCES

Akutsu,T., Kuhara,S., Maruyama,O. and Miyano,S. (1998) Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Mathematics (SODA 98)*. pp. 695–702.

Akutsu,T., Miyano,S. and Kuhara,S. (1999) Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of the 1999 Pacific Symposium in Biocomputing (PSB 99)*. pp. 17–28.

Arkin,A., Ross,J. and McAdams,H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda infected escherichia coli cells. *Genetics*, **149**, 1275–1279.

Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. In *Proceedings of the 1999 Pacific Symposioum in Biocomputing (PSB 99)*. pp. 29–40.

Cover,T.M. and Thomas,J.A. (1991) *Elements of Information Theory*. John Wiley and Sons, Inc..

Derisi,J., Iyer,V. and Brown,P. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **282**, 699–705.

Dhaeseleer,P., Wen,X., Fuhrman,S. and Somogyi,R. (1999) Linear modeling of mRNA expression levels during CNS development and injury. In *Proceedings of the 1999 Pacific Symposioum in Biocomputing (PSB 99)*. pp. 41–52.

Eisen,M., Spellman,P., Brown,P. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using Bayesian networks to analyze expression data. *J. Comp. Biol.*, **7**, 601–620.

Hanisch,D., Zien,A., Zimmer,R. and Lengauer,T. (2002) Co-clustering of biological networks and gene expression data. *Bioinformatics*, **18**, 145–154.

Heckerman,D., Geiger,D. and Chickering,D. (1995) Learning Baysian networks: the combination of knowledge and statistical data. *Technical Report MSR-TR-94-09*. Microsoft research.

Ideker,T., Thorsson,V. and Karp,R. (2000) Discovery of regulatory interaction through perturbation: inference and experimental design. In *Proceedings of the 2000 Pacific Symposioum in Biocomputing (PSB 00)*. pp. 305–316.

Ideker,T., Thorsson,V., Ranish,J.A., Christmas,R., Buhler,J., Eng,J.K., Bumgarner,R., Goodlett,D.R., Aebersold,R. and Hood,L. (2001) Integrated genomic and proteomic analyses of systematically perturbed metabolic network. *Science*, **292**, 929–933.

Ideker,T., Ozier,O., Schwikowski,B. and Siegel,A.F. (2002) Discovering regulatory and signaling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–S240.

Jones,E.W., Pringle,J.R. and Broach,J.R. (eds) (1992) *The Molecular and Cellular Biology of the Yeast Saccharomyces: Gene Expression*. Cold Spring Harbor Laboratory Press.

Kauffman,S. (1974) The large scale structure and dynamics of gene control circuits: an ensemble approach. *J. Theor. Biol.*, **44**, 167–190.

Lehmann,E. and D'abrera,H. (1975) *Nonparametrics*. Holded-day inc, McGrow-Hill, NY.

Liang,S., Fuhrman,S. and Somogyi,R. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of the 1998 Pacific Symposioum in Biocomputing (PSB 98)*. pp. 18–29.

Lockhart,D., Dong,H., Byrne,M., Follettie,M., Gallo,M., Chee,M., Mittmann,M., Wang,C., Kobayashi,M., Horton,H. *et al.* (1996) DNA expression monitoring by hybridization of high density oligonucleotide arrays. *Nat. Biotechnol.*, **14**, 1675–1680.

Mannervik,M., Nibu,Y., Zhang,H. and Levine,M. (1999) Transcriptional coregulators in development. *Science*, **284**, 606–609.

Neidhardt,F.C. (ed) (1996) *Escherichia coli and Salmonella: Cellular and Molecular Biology*. ASM Press.

Pe'er,D., Regev,A. and Tanay,A. (2002) Minreg: inferring an active regulator set. *Bioinformatics*, **18**, S258–S267.

Roberts,C., Nelson,B., Marton,M., Stoughton,R., Meyer,M.R., Bennett,H.Y., Dai,H., Walker,W., Hughes,T. *et al.* (2000) Signaling and circuitry of multiple MAPK pathways revealing by a matrix of global gene expression profile. *Science*, **287**, 873–880.

Segal,E., Taskar,B., Gasch,A.N. and Friedman,D.K. (2001) Rich probabilistic models for gene expression. *Bioinformatics*, **17**, 243–252.

Sharan,R., Elkon,R. and Shamir,R. (2002) Cluster analysis and its applications to gene expression data. In Mewes,H., Seidel,H. and Weiss,B. (eds), *Bioinformatics and Genome Analysis*. Springer, pp. 83–108.

Sokal,R.R. and Rohlf,F.J. (1995) *Biometry: The Principles and Practice of Statistics in Biological Research*. W.H. Freeman and company.

Tanay,A. and Shamir,R. (2001) Computational expansion of genetic networks. *Bioinformatics*, **17**, S270–S278.

Thieffry,D. and Thomas,R. (1998) Qualitative analysis of gene networks. In *Proceedings of the 1998 Pacific Symposioum in Biocomputing (PSB 98)*. pp. 77–88.

Wilf,H. (1994) *GeneratingFunctionology*. Academic Press.

Wilks,S.S. (1938) The large sample distribution of the likelihood ratio for testing composite hypotheses. *Ann. Math. Stat.*, **9**, 60–62.

APPENDIX A: PROPERTIES OF CHAIN FUNCTIONS

In this section, we prove some properties of chain functions. We study the problem of counting the exact

number of chain functions with n variables, and provide a tight asymptotic bound. We use the same terminology as in the Chain Functions Section.

LEMMA 1. *Every regulator in a chain function is a direct effector of the regulatee.*

PROOF. To show that g_i is a direct effector of g_0, consider a combination of states for $g_n \dots g_{i+1}$ which creates a positive influence on g_i (for example, all the variables with A(R) control bit have the state 1(0)). If the states of $g_{i-1}, \dots, g_1$ are all 1s, then the state of g_0 is dependent on the state of g_i, and thus g_i is a direct effector of g_0. □

LEMMA 2. *The layers partition the regulators into a minimal number of classes.*

PROOF. We shall show first that the regulators in a layer form a class. Then, we shall show that a successive pair of regulators with the control pattern RA or RR must be in different classes, and thus we must truncate the classes after each R.

We start with the first claim. A consecutive pair of regulators inside a layer always has the pattern AA or AR. Exchanging the order of the two regulators might influence the state of g_0 only if the two regulators have different states: in the AA control pattern, the state patterns 10 and 01 both yield negative influence, irrespective of the previous influences. Likewise, in the AR control pattern, the state patterns 10 and 01 both yield positive influence. Thus, any two consecutive regulators inside a layer are exchangeable. Therefore, any permutation of regulators in a layer might be reached by a series of successive pair exchanges without changing the function.

Next, we show the second claim. In the RA control pattern, the state pattern 10 yields negative influence while the state pattern 01 yields positive influence. Likewise, in the RR control pattern, the state pattern 10 yields positive influence while the state pattern 01 yield negative influence. Thus, such pairs are unexchangeable. □

In the rest of the appendix, we study the problem of chain functions counting. Define the *composition* of a layer as the subset of regulators out of $g_n, \dots, g_1$, which correspond to the layer.

LEMMA 3. *A chain function is uniquely determined by the sizes, order and composition of its layers, and the type of pattern in the first layer.*

PROOF. To prove the lemma, we show that any change in the number, order or composition of the layers, or the type of the first layer, is not function preserving. First, we prove that different types of the first layers cannot correspond to the same function: Given the state pattern $000\dots0$ for all regulators, if the first layer is of type A, it has negative influence and the state of the regulatee is 0. If it is of type R, that state is 1, so the function value is changed.

Next we prove that any change in the number, order or composition of layers is not function preserving. Let f'^g and f^g be two chain functions whose number, order or composition of layers is different. The layers of f'^g are denoted by $l'_p, \dots, l'_1$, and the layers of f^g are denoted by $l_q, \dots, l_1$. We denote by l_x and l'_x the first (least indexed) layers whose composition differs, so that the layers $l'_{x-1}, \dots, l'_1$ are identical to the layers $l_{x-1}, \dots, l_1$, and l_x is different from l'_x. Such layers l_x and l'_x exist by our assumptions. Suppose, w.l.o.g, that l'_x contains a variable v which is not included in l_x. Assume that l_x and l'_x have the same type R (The proof for the other layer type is similar). Consider the following state pattern: All variables in layers $l_x, \dots l_1$ are in state 1 and all the rest (including variables that appear in only one of the functions) are in state 0. Thus, l_x is positively influenced by layer l_{x+1} and is negatively influencing its successor. However, using the same state pattern, l'_x contains the variable v which has state 0. Thus, l'_x has positive influence on l'_{x-1}. Since layers $l'_{x-1}, \dots, l'_1$ have the same composition as $l_{x-1}, \dots, l_1$ and the state pattern in both is all 1-s, the final function value is changed. □

We now count the total number of chain functions with n variables. Let S^n_k be the number of partitions of n variables into exactly k nonempty sets. S^n_k may be computed recursively by the formula $S^n_k = kS^{n-1}_k + S^{n-1}_{k-1}$, where $S^x_1 = 1$, $S^x_0 = 0$ and $S^x_y = 0$ for $y > x$. In each step we add a variable to one of the k existing sets, or we put the variable in the new set. Thus, the number of partitions of n variables into any number of ordered nonempty sets is $\tilde{b}(n) = \sum_{k=1}^{n} k! \cdot S^n_k$. $\tilde{b}(n)$ is known as an *ordered Bell number*, which is asymptotically $(1 + O(1)) \cdot \frac{n!}{2} \cdot (\log_2 e)^{n+1}$ (Wilf, 1994, p. 175–176). For each partition of the variables, there are two possible types of first layer. Thus we conclude:

THEOREM 4. *The number of chain functions with n control variables is* $2 \cdot \tilde{b}(n)$.

Hence the number is $\Theta(n! \cdot (\log_2 e)^{n+1})$. For example, for $n = 2$, there are $2 \cdot \tilde{b}(2) = 6$ different functions. Indicating the chain functions as $f_{c_2,c_1}(g_2, g_1)$, these are $f_{R,R}(x, y)$, $f_{R,R}(y, x)$, $f_{R,A}(x, y)$, $f_{R,A}(y, x)$, $f_{A,A}(x, y)$ (equivalent to $f_{A,A}(y, x)$, since AA is one layer), and $f_{A,R}(x, y)$ (equivalent to $f_{A,R}(y, x)$, since AR is one layer).

SIAM J. DISCRETE MATH.
Vol. 20, No. 3, pp. 727–740

RECONSTRUCTING CHAIN FUNCTIONS IN GENETIC NETWORKS*

IRIT GAT-VIKS†, RICHARD M. KARP‡, RON SHAMIR†, AND RODED SHARAN†

Abstract. The following problems arise in the analysis of biological networks: We have a boolean function of n variables, each of which has some default value. An *experiment* fixes the values of any subset of the variables, the remaining variables assume their default values, and the function value is the result of the experiment. How many experiments are needed to determine (reconstruct) the function? How many experiments that involve fixing at most q values are needed? What are the answers to these questions when an unknown subset of the variables are actually involved in the function? In the biological context, the variables are genes and the values are gene expression intensities. An experiment measures the gene levels under conditions that perturb the values of a subset of the genes. The goal is to reconstruct the particular logic (regulation function) by which a subset of the genes together regulate one target gene, using few experiments that involve minor perturbations. We study these questions under the assumption that all functions belong to a biologically motivated set of so-called chain functions. We give optimal reconstruction schemes for several scenarios and show their application in reconstructing the regulation of galactose utilization in yeast.

Key words. network reconstruction, experimental design

AMS subject classifications. 90B10, 62K99, 06E30

DOI. 10.1137/S089548010444376X

1. Introduction.

In this paper we study the problem of function reconstruction. We have a set of N boolean variables. Each variable has a default value, and an *experiment* can change (fix to 0 or 1) its value. The *order* of an experiment is the number of variables fixed during the experiment. The value of one variable of interest (the output) is determined by a boolean function of n other variables. The *output* of an experiment is the value of the function, where all fixed variables attain their respective values and the rest attain their default values. The problem of *function reconstruction* is to determine this function using a minimum number of experiments of the smallest possible order.

The motivation to studying the problem arises in molecular biology: The regulation of biological entities is key to cellular function. The genes are expressed (transcribed) into mRNAs, which are translated into proteins. The regulatory factors which control (regulate) gene expression are themselves protein products of other genes. The result is a complex network of regulatory relations among genes. A *genetic network* consists of a set of variables that correspond to *genes*, attaining real values, called *states*. The state of a gene indicates the discretized expression level of the gene. A gene may be *regulated* by several other genes, implying that its state

*Received by the editors May 16, 2004; accepted for publication (in revised form) January 24, 2006; published electronically October 4, 2006. The work of the first author was supported by a Colton fellowship. The second and third authors were supported by a grant from the US-Israel Binational Science Foundation (BSF). The fourth author was supported by a Fulbright grant and by NSF ITR grant CCR-0121555. A preliminary version of this paper appeared in *Proceedings of the Ninth Pacific Symposium on Biocomputing* [10].

http://www.siam.org/journals/sidma/20-3/44376.html

†School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (iritg@tau.ac.il, rshamir@tau.ac.il, roded@tau.ac.il).

‡International Computer Science Institute, 1947 Center St., Berkeley, CA 94704 (karp@icsi.berkeley.edu).

is a function of the states of its regulating genes, or its *regulators*. An experiment involves *perturbations* such as knocking out certain genes (fixing their states to some low value) or overexpressing them (fixing their states to some high value) and measuring the expression levels of all other genes. The measurement of gene expression levels is facilitated by high throughput technologies, such as DNA microarrays (e.g., [6]). The order of an experiment is the number of genes that are perturbed. In order to reconstruct the regulatory relations among genes, we need to infer the set of genes that cooperate in the regulation of a given gene and the particular logical function by which this regulation is determined. This paper studies the number and order of experiments that are needed in order to infer the regulatory function that governs a specific gene.

A key obstacle in the inference of regulation relations is the large number of possible solutions and, consequently, the unrealistically large amount of data needed to identify the right one. A common and simple model for genetic networks is the *boolean* model, in which the state of a gene is 0 (off) or 1 (on). The boolean assumption is a drastic simplification of real biology, yet it captures important features of biological systems and was frequently used in previous studies [16].

There is a large body of previous work on learning boolean functions from a random sample of their output values (see [3] for a review). Those studies focus on devising efficient probably approximately correct (PAC) learning algorithms for subclasses of boolean functions using a polynomial-size sample. Another body of work is devoted to exact learning of certain classes of boolean functions using a polynomial number of queries (see, e.g., [4] and references thereof). For the specific problem of exact boolean function reconstruction in a genetic network, Akutsu et al. [1] have shown that the number of experiments (or queries) that are needed for reconstructing a function of N genes is prohibitive: The lower and upper bounds on the number of experiments of order $N-1$ that are needed are $\Omega(2^{N-1})$ and $O(N \cdot 2^{N-1})$, respectively. When the function involves only d regulators, the number of required experiments of order d is still $\Omega(N^d)$ and $O(N^{2d})$, respectively [1].

The inherent complexity of this problem led researchers to seek ways around this problem. Ideker, Thorson, and Karp [16] studied how to dynamically design experiments so as to maximize the amount of information extracted. Friedman et al. [8] used Bayesian networks to reveal parts of the genetic network that are strongly supported by the data. Tanay and Shamir [24] suggested a method of expanding a known network core using expression data. Several studies used prior knowledge about the network structure, or restrictive models of the structure, in order to identify relevant processes in gene expression data [12, 15, 23, 22].

Recently, a biologically motivated, boolean model of regulation relations based on *chain functions* was suggested in order to cope with the problem of function reconstruction in biological context [9]. In a chain function, the state of the regulated gene depends on the influence of its direct regulator, whose activity may in turn depend on the influence of another regulator, and so on, in a chain of dependencies (we defer formal definitions to the next section). The class of chain functions has several important advantages [9]: These functions reflect common biological regulation behavior, so many real biological regulatory relations can be elucidated using them (examples include the SOS response mechanism in *E. coli* [21] and galactose utilization in yeast [18]). Moreover, by restricting consideration to chain functions, the number of candidate functions drops from double exponential to single exponential only.

In this paper we study several computational problems arising when wishing to

reconstruct chain functions using a minimum number of experiments of the smallest possible order. We address both the question of finding the set of regulators of a chain function, which is typically much smaller than the entire set of genes, and the question of reconstructing the function given its regulators. We give optimal reconstruction schemes for several scenarios and show their application on real data. Our analysis focuses on the theoretical complexity of reconstructing regulation relations (number and order of experiments), assuming that experiments provide accurate results and that the target function can be studied in isolation from the rest of the genetic network.

The paper is organized as follows: Section 2 contains basic definitions related to chain functions. In section 3 we give worst-case and average-case analyses of the number of experiments needed in order to reconstruct a chain function. Both low-order and high-order experimental settings are considered. In section 4 we study the reconstruction of composite regulation functions that combine several chains. Finally, in section 5 we describe a biological application of our analysis to reconstruct the regulation mechanism of galactose utilization in yeast.

2. Chain functions. Chain functions were introduced by Gat-Viks and Shamir [9]. In the following we define these functions and describe their main properties. Our presentation differs from the original one to allow succinct description of the reconstruction schemes in later sections.

Variables, regulators and states. Let U denote the set of all variables in a network, where $|U| = N + 1$. These variables correspond to genes, mRNAs, proteins, or metabolites. Each variable may attain one of two *states*: 1 or 0. The state of gene g, denoted by $state(g)$, indicates the discretized expression level of the gene. A variable normally attains its *wild-type* state, but perturbations such as gene knockouts may change its state. We say that a variable $g_0 \in U$ is *regulated by* a set $S = \{g_1, \ldots, g_n\} \subset U$ if $state(g_0) = f^{g_0}(state(g_n), \ldots, state(g_1))$ and S is a minimal set with that property. In that case we say that S is the *regulator set* of g_0, and g_0 is called the *regulatee*. Associated with each regulator g_i is a binary constant y_i which dictates the *control* property of g_i. If $y_i = 0$ then g_i is an *activator*; otherwise g_i is a *repressor*. This is an intrinsic property of the regulator and is not subject to change. The *control pattern* of f^{g_0} is the binary vector $(y_n, \ldots, y_1)$.

Given a certain order $g_n, \ldots, g_1$ of the regulators, we call g_i a *predecessor* of g_j for $i > j$ and a *successor* of g_k for $i < k$. We also say that g_i is to the *left* of g_j and to the *right* of g_k. Each regulator transmits a signal to its immediate successor, and this chain of events enables a signal to propagate from g_n to g_0 in a manner defined by a chain function (see Figure 1, top part).

Chain function definition. The chain function model assumes that the functional relations are deterministic. The chain function f^{g_0} on the regulators $g_n, \ldots, g_1$ determines the state of the regulatee g_0.

The function f^{g_0} can be defined using two n-long boolean vectors attributing *activity* and *influence* to each g_i. Let $a(g_i)$ denote the activity of g_i, and let $infl(g_i)$ denote the influence signal from g_i to g_{i-1}. The definitions of activity and influence on the other regulators are recursive: The influence on g_n is always 1. g_i is active ($a(g_i) = 1$) iff it exists ($state(g_i) = 1$) and it receives a positive influence from its predecessor ($infl(g_{i+1}) = 1$). The influence $infl(g_i)$ transmitted from g_i to g_{i-1} is a xor ($\oplus$) of $a(g_i)$ and y_i: $infl(g_i)$ is 1 if g_i is an activator and is itself activated or if g_i is a repressor and is not activated (so that it fails to repress g_{i-1}). Formally,

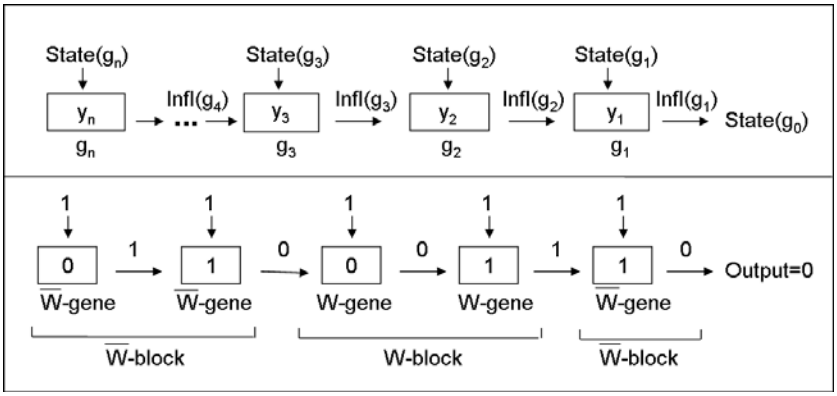


FIG. 1. *The chain function model. Top: A chain function model. Bottom: An illustration of a chain function with five regulators.* g_1, g_2, g_4 *are repressors, and* g_3, g_5 *are activators. The state of all regulators is* 1. *Influences are indicated on the horizontal arrows. Regulator types and blocks are indicated below.*

$$a(g_i) \;=\; infl(g_{i+1}) \;\wedge\; state(g_i), \tag{1}$$

$$infl(g_i) \;=\; y_i \oplus a(g_i). \tag{2}$$

Finally, the state of the regulatee g_0 is simply the influence of g_1. We define the *output* of f^{g_0} to be $state(g_0)$.

A chain function is uniquely determined by its set of regulators, their order, and the control pattern. For example, if g_0 is regulated by (g_3, g_2, g_1) via a chain function with control pattern 010, then $f(1,1,1) = 0$ and $f(0,1,1) = 1$.

3. Reconstruction of chain functions. In this section we study the question of uniquely determining the chain function which operates on a known regulatee, using a minimum number of experiments. We assume throughout that all variable states in wild type are known. We further assume that all regulator states in wild type are 1, except possibly g_n. The latter assumption is motivated by the observation that in many biological examples, all regulators are expressed in wild type, and the state of the regulatee is determined by the presence or absence of a metabolite g_n. (Examples include the Trp, lac, and araBAD operons in *E. coli* [21], the regulation of galactose utilization [18] in yeast, and human MAPK cascades [17]).

An *experiment* is defined by a set of variables that are externally perturbed (knocked-out or overexpressed). The states of the perturbed variables are thus fixed, and the states of all nonperturbed regulators are assumed to remain at the wild-type values. The state of the regulatee is determined by the chain function. The *order* of an experiment is the number of externally perturbed variables in it.

Our reconstruction algorithms are based on performing various experiments and observing their effect on the state of the regulatee. The algorithms implicitly assume that the regulation function is indeed a chain function and do not explicitly test this property.

We now devise a simple set of equations that characterize the output of a chain function as a function of the control pattern and the states of the regulators, both

in the wild-type state and in states produced by perturbing some regulators. These equations are the foundation of all the subsequent reconstruction schemes:

PROPOSITION 1. *Let f be a chain function on $g_n, \ldots, g_1$. If $state(g_i) = 1$ for $1 \le i < n$, then $state(g_0) = state(g_n) \oplus (\oplus_{i=1}^{n} y_i)$. For any other state vector, if the least index of a state-0 regulator is $j \le n$, then $f^{g_0}(g_n, \ldots, g_1) = \oplus_{i=1}^{j} y_i$.*

Proof. By definition, $a(g_n) = state(g_n)$. For $i < n$, $state(g_i) = 1$ implies that $a(g_i) = a(g_{i+1}) \oplus y_{i+1}$. It follows by induction that $state(g_0) = state(g_n) \oplus (\oplus_{i=1}^{n} y_i)$. Similarly, if $state(g_j) = 0$ and $state(g_i) = 1$ for all $i < j$, it follows by induction that $f^{g_0}(g_n, \ldots, g_1) = \oplus_{i=1}^{j} y_i$. ∎

Under the above assumptions on regulator states, a chain function can be viewed as a series of inversion and identity gates, whose input is the state of g_n. Each identity gate corresponds to an activator, whose output is equal to its input. Each inversion gate corresponds to a repressor, whose output is opposite to its input. The output of the last gate in the chain is the state of the regulatee.

3.1. Types and blocks. A *perturbation* is an experiment that changes the state of a variable to the opposite of its state in wild type. By our assumption on the regulator states in wild type (all regulator states in wild type are 1, except possibly g_n), the perturbation of a regulator in $\{g_{n-1}, \ldots, g_1\}$ is a knockout. For $S \subseteq U$, an *S-perturbation* is an experiment in which the states of all the variables in S are perturbed.

Let w be $state(g_0)$ in wild type. Let $\bar{w}$ be the opposite state. For the reconstruction, we first classify the variables in U into two *types*: W and $\bar{W}$ (see Figure 1, bottom part). A variable is in W ($\bar{W}$) if its perturbation produces output w ($\bar{w}$). Typically, the majority of the genes have type W, since in particular all the genes that are not part of the chain function are such. By Proposition 1 we have $g_n \in \bar{W}$, and $g_{n-1} \in W$ iff $state(g_n) \oplus y_n = 0$. We call a gene that belongs to W ($\bar{W}$) a *W-gene* (*$\bar{W}$-gene*). Similarly, we call a regulator of type W ($\bar{W}$) a *W-regulator* (*$\bar{W}$-regulator*). For a given gene, we call a successor of type W ($\bar{W}$) of that gene a *W-successor* (*$\bar{W}$-successor*).

The type of a gene can be determined by a single perturbation of the gene. Such an experiment will be referred to as a *typing experiment* throughout.

COROLLARY 2. *Given an ordered set of regulators $g_n, \ldots, g_1$, their control pattern can be reconstructed using $n - 1$ typing experiments.*

Proof. Perform typing experiments for $g_1, \ldots, g_{n-1}$ (by definition $g_n \in \bar{W}$). By Proposition 1, for every $1 < i < n$, $y_i = 1$ iff the types of g_i and g_{i-1} differ. Also, $y_n = 1$ iff either $state(g_n) = 0$ and the types of g_n, g_{n-1} are equal, or $state(g_n) = 1$ and the two types differ. Finally, we can use Proposition 1 to deduce y_1. ∎

Any control pattern $(y_n, \ldots, y_1)$ may be separated into *blocks* of consecutive regulators by truncating the control pattern after each 1. The first block (rightmost, ending at g_1) has two possible forms: $0 \ldots 0$ or $0 \ldots 01$. All other blocks are of the form $0 \ldots 01$, so the right boundary of a block corresponds to a regulator g_j with $y_j = 1$, and any other regulator g_i in the block has $y_i = 0$.

LEMMA 3. *Each block contains regulators of a single type, and two adjacent blocks contain regulators of opposite types.*

The proof follows from the fact that the type of g_i, $i < n$ differs from the type of g_{i-1} iff $y_i = 1$. Thus, we can refer to a block as either a *W-block* or a *$\bar{W}$-block*, and the two types of blocks alternate. For convenience, we shall refer to g_n as forming a $\bar{W}$-block of its own.

3.2. Reconstructing the regulator set and the function. Consider a chain function with control pattern $(y_n, \ldots, y_1)$ and let $g_j, \ldots, g_i$ be a block. Then $infl(g_i) = [infl(g_{j+1}) \wedge (\bigwedge_{h=i}^{j} state(g_h))] \oplus y_i$. Thus, the effect of the block on the function is determined by the boolean variable $infl(g_{j+1})$, by the control pattern, and by the conjunction of the states of its regulators. Since this conjunction is independent of the order of occurrence of these genes, no experiment based on perturbing the states of the genes can determine the order of the genes within the block. In view of this limitation, we shall aim to find the equivalence class of chain functions as detectable by perturbation experiments, i.e., our goal is to reconstruct the control pattern, the set of genes within each block (but not their order), and the ordering of the blocks. Correspondingly, in the following we will use the term *successor* of a gene to denote a regulator that succeeds that gene in the chain and is not a member of its block. For convenience, we shall refer to gene (in fact, W-genes) that are not regulators of g_0 as predecessors of g_n.

The above discussion implies that once we have typed each gene, it remains to determine, for each pair consisting of a W-gene and a $\bar{W}$-gene, which one precedes the other in the chain. Let k_W and $k_{\bar{W}}$ denote the number of regulators of type W and $\bar{W}$, respectively. Note that $k_W + k_{\bar{W}} = n \leq N$, and in fact, typically, $n \ll N$ as $k_W \ll |W|$.

Suppose we perform a $\{i, k\}$-perturbation with $g_i \in W$ and $g_k \in \bar{W}$. If the result is w, then g_k precedes g_i. Otherwise, g_i precedes g_k. A 2-order experiment for determining the relative order of a W-gene and a $\bar{W}$-gene will be called a *comparison* throughout.

PROPOSITION 4. *Given the set of regulators of a chain function and their types, $k_W k_{\bar{W}}$ comparisons are necessary and sufficient to reconstruct the function.*

Proof. The upper bound follows by comparing every W-regulator with every $\bar{W}$-regulator. The lower bound follows from the fact that, in the special case where every $\bar{W}$-regulator precedes every W-regulator, no set of comparisons can determine the relative order of a given pair consisting of a W-regulator and a $\bar{W}$-regulator, unless it includes a direct comparison between the pair. Therefore, all such comparisons must be performed. □

Note that the problem of reconstructing a chain function by comparisons, once the regulators have been typed, can be viewed as a sorting problem: The input is a list of n elements of two types, such that the set of elements of each type consists of several equivalence classes, and there is a linear order of all these classes. The objective is to find the equivalence classes and their order, using only queries that compare two elements of distinct types. In the special case that each equivalence class consists of one element, the problem is related[1] to the well-studied problem of matching nuts and bolts [2] and has an optimal $\Theta(n \log n)$ deterministic solution [19].

We now turn to the question of reconstructing a chain function without prior knowledge of the identity of its regulators. The discussion above suggests a way to solve the problem: First, we find the gene types using N typing experiments. Next, we reconstruct the block structure by performing all possible comparisons between a W-gene and a $\bar{W}$-gene.

A more efficient reconstruction is possible when g_n is known. This is often the case when the chain function models a signal transduction pathway, where g_n represents

[1] The difference between the problem of matching nuts and bolts and our problem is that in our case we have strict linear order among all the elements and there is no notion of matching between W-regulators and $\bar{W}$-regulators.

a known stimulator of the corresponding biological response. If g_n is known, then since $g_n \in \bar{W}$, all W-regulators can be identified by comparing every W-gene with g_n, using a total of $N - k_{\bar{W}}$ comparisons. Since every $\bar{W}$-gene is a regulator, these experiments are sufficient to identify all the regulators, and we can apply Proposition 4 to complete the reconstruction in $N - k_{\bar{W}} + k_W(k_{\bar{W}} - 1)$ comparisons. In summary, we have the following proposition.

PROPOSITION 5. *A chain function can be reconstructed using at most N typing experiments and $k_{\bar{W}}(N - k_{\bar{W}})$ comparisons. Given g_n, a chain function can be reconstructed using at most $N - 1$ typing experiments and $N - n + k_W k_{\bar{W}}$ comparisons.*

We can prove a matching lower bound by generalizing the argument in Proposition 4.

PROPOSITION 6. *At least $k_{\bar{W}}(N - k_{\bar{W}})$ comparisons are necessary to reconstruct a chain function.*

Proof. Consider the case where all $\bar{W}$-regulators precede the W-regulators. In this case, no set of comparisons can determine the relative order of a given pair consisting of a W-gene and a $\bar{W}$-gene unless it includes a direct comparison between the pair. Therefore, all such comparisons must be performed. □

Propositions 4 and 5 provide a worst-case analysis. Next, we describe another reconstruction algorithm, whose *expected* number of required experiments is lower. The analysis of the running time is similar to that of quick-sort (cf. [5]) and assumes that the chain to be reconstructed has $\bar{W}$-blocks of bounded size. Denote by D_g the set of W-successors of $g \in \bar{W}$ in f.

PROPOSITION 7. *A chain function with $\bar{W}$-blocks of size bounded by d can be reconstructed using N typing experiments and an expected number of $O(Nd \log k_{\bar{W}} + k_W k_{\bar{W}})$ comparisons.*

Proof.

Algorithm: First, we perform N typing experiments. Next, we apply a randomized scheme to reconstruct the chain: Each time we pick a gene $g \in \bar{W}$ at random, find its successors and their order, and remove g and all its successors from further consideration. We stop when no $\bar{W}$ genes are left. In order to find the successors of g, we first identify the members of D_g using at most $N - k_{\bar{W}}$ comparisons. Using D_g, we then reconstruct the part of the chain that spans g and its successors by at most $|D_g|(k_{\bar{W}} - 1)$ comparisons, as in Proposition 4.

Complexity: The set of comparisons can be divided into two parts: those that are required to identify the sets D_g and those required to reconstruct the chain parts induced by these sets. For the latter, at most $k_W k_{\bar{W}}$ comparisons are needed in total, since every pair consisting of a W-regulator and a $\bar{W}$-regulator is compared at most once. Thus, it suffices to compute the expectation of the first part. Let $T(x)$ be this expectation, given that the current $\bar{W}$ set (i.e., the set of $\bar{W}$-genes that were not removed in previous iterations) contains x elements, where $T(0) = 0$. For $x \geq 1$, with probability $\frac{1}{x}$ the qth rightmost element of $\bar{W}$ is chosen in the current iteration. Hence, $T(x) \leq \frac{1}{x} \sum_{q=1}^{x} (d(N - k_{\bar{W}}) + T(x - q))$. By induction, $T(x) \leq d(N - k_{\bar{W}})(\log x + 1)$. Substituting $x = k_{\bar{W}}$, we obtain the required bound. □

The expected number of experiments improves over the upper bound of Proposition 5 for $d < k_{\bar{W}}$, which is the case in many real biological regulations, e.g., the filamentous-invasion pathway ($n = 9$, $k_{\bar{W}} = 2$, and $d = 1$, illustrated in [11, Figure 3]), and the HOG signaling pathway ($n = 6$, $k_{\bar{W}} = 3$, and $d = 2$ [13]) in yeast.

3.3. Using high-order experiments. In this section we show how to improve the above results when using experiments of order $q > 2$. The results in this section

are mainly of theoretical interest, since high-order experiments may not be practical.

PROPOSITION 8. *Given the set of n regulators of a chain function, the function can be reconstructed using $O(\frac{n^2}{q}\log q)$ experiments of order at most $q \le n$. This is optimal up to constant factors for $q = \Theta(n)$.*

Proof. The number of possible chain functions with n regulators is $\Theta((\log_2 e)^{n+1}n!)$ [9]. Since each experiment provides one bit of information, the information lower bound is $\Omega(n \log n)$ experiments.

Suppose at first that $q = n$. Let n_i be the number of regulators in block i, where blocks are indexed in right-to-left order. Our reconstruction algorithm is as follows: First, we perform n typing experiments. Next, we identify the type of the first block using one experiment of order n, in which all regulators are perturbed (this way we perturb also the genes in the first block, and thus its type is identical to the output). We proceed to reconstruct the blocks one by one, according to their order along the chain. Note that the type of each block is now known, since the two types alternate. Suppose we have already reconstructed blocks $1, \ldots, i-1$. For reconstructing the ith block we only consider the set of regulators that do not belong to the first $i-1$ blocks. Out of this set, let A be the subset of regulators that have the same type as block i, and let B be the subset of regulators of the opposite type. In order to identify the members of the ith block we use a binary-search-like procedure: We divide A into two halves. For each half we perform a perturbation that includes that half and all regulators in B. If the result is the type of block i, we continue recursively with that half. Otherwise, we discard it. The search requires $O(n_i \log n)$ experiments. Thus, altogether we perform $O(n \log n)$ experiments.

When $q < n$, we use the above algorithm as a component in our reconstruction scheme, allowing us to reconstruct a subchain of size q within a chain of size n using $O(q \log q)$ experiments of order at most q. Our reconstruction scheme is based on Proposition 4, which shows that for reconstruction it suffices to compare every W-regulator with every $\bar{W}$-regulator. To this end we form $O(\frac{n^2}{q^2})$ regulator subsets, each of size at most q, such that every pair consisting of a W-regulator and a $\bar{W}$-regulator appears in one of the subsets. To compute these subsets we form a $k_W \times k_{\bar{W}}$ matrix, whose entries are in 1-1 correspondence with $(W, \bar{W})$-regulator pairs. We then cover this matrix using $O(\frac{k_W k_{\bar{W}}}{q^2})$ disjoint submatrices of dimension at most $\lfloor q/2 \rfloor \times \lceil q/2 \rceil$, each identifying a regulator subset of the required size.

Next, we reconstruct the subchain of size q associated with each subset using $O(q \log q)$ experiments of order at most q. After this process, each $(W, \bar{W})$-regulator pair appears in one of the subchains, and thus its relative order has been determined. This is sufficient in order to computationally reconstruct the chain (as in Proposition 4). Altogether we use $O(\frac{k_W k_{\bar{W}}}{q} \log q) = O(\frac{n^2}{q} \log q)$ experiments for reconstructing the chain from its regulators. □

We now provide a reconstruction scheme for the case that the set of regulators is not known. Let f be a chain function. For a gene $g \in \bar{W}$, denote as before by D_g its set of W-successors in f. A building block in our reconstruction scheme is a method to efficiently identify the members of D_g using $O(|D_g| \log q + N/q)$ experiments of order at most q. The process is as follows: We partition the W-genes into $\lceil \frac{N}{q-1} \rceil$ subsets of size at most $q-1$. For each subset R we test whether it contains some successor of g using an $(R \cup \{g\})$-perturbation, in which g and the subset members are perturbed. If as a result of the perturbation the output changes to w, then at least one of the members in R succeeds g. In this case we use standard binary search to identify all the m successors in R by performing additional $O(m \log q)$ experiments of order at

most $(\lfloor q/2 \rfloor + 1)$. Otherwise, all the subset members precede g and we discard R. Each of the successors of g is discovered exactly once, which gives the required bound.

PROPOSITION 9. *For $q \leq n$, a chain function can be reconstructed using $O(nN/q + n^2 \log q/q)$ experiments of order at most q. For $q > n$, $O(N + n \log q)$ experiments of order at most q are sufficient.*

Proof. The reconstruction is done in three stages. First, we perform N typing experiments. Second, we discover all W-regulators as follows: For each regulator $b \in \bar{W}$ we use the scheme described above to identify its successors in W, and remove them from further consideration. Each W-regulator is discovered exactly once and, thus, we need $O(k_{\bar{W}} N/q + k_W \log q)$ experiments of order at most q altogether. Last, we reconstruct the chain, given the regulators and their types, in $O(n^2 \log t/t)$ experiments, using the method given in Proposition 8, where $t = \min\{q, n\}$. In total $O(N + k_{\bar{W}} N/q + k_W \log q + n^2 \log t/t)$ experiments are used. □

A lower bound on the number of experiments that are required is given in the following proposition.

PROPOSITION 10. $\Omega(\max\{N/q, nN/q^2, n \log N\})$ *experiments of order at most q are necessary to reconstruct a chain function.*

Proof. We give three different lower bounds, whose union yields the required result. First, $\Omega(N/q)$ experiments are required to identify at least one $\bar{W}$-regulator. Second, $\Omega(nN/q^2)$ experiments are required to cover every pair of a W- and a $\bar{W}$-gene. Third, the number of possible chain functions is $\Theta(\binom{N}{n}(\log_2 e)^{n+1} n!)$ [9]. Hence, the information theoretic lower bound on the reconstruction is $\Omega(n \log N)$. □

Finally, we give an optimal reconstruction scheme when g_n is known and $q = \lceil N/2 \rceil + 1$.

PROPOSITION 11. *In case g_n is known, there is an optimal reconstruction scheme that uses $\Theta(n \log N)$ experiments of order at most $\lceil N/2 \rceil + 1$.*

Proof. We perform the reconstruction in two stages. In the first stage we discover the set of regulators and their types. In the second stage we apply Proposition 8 to reconstruct the chain function. To discover the set of regulators we perform a binary-search-like process as follows: We partition all variables excluding g_n and g_0 into two halves, H_1 and H_2. For $i = 1, 2$ we apply an $H_i \cup \{g_n\}$-perturbation. Since g_n is perturbed, all nonregulator effects are masked, and we get the result w iff H_i contains some W-regulators. Therefore, for each set that gives the results w, we continue recursively until we reach single genes. In this way we have identified a subset T of the W-regulators, including all those in the first (rightmost) block. We now repeat the recursive process on $U \setminus (T \cup g_n \cup g_0)$, but this time do not include g_n in the perturbations. This process identifies a subset T' of the $\bar{W}$ regulators, including the first $\bar{W}$-block. By repeating these two recursive processes (with and without including g_n in the perturbations) we eventually identify all regulators. The total effort is $O(n \log N)$ since each path that identifies one of the n regulators is a binary search in N variables and thus takes $O(\log N)$ experiments. □

4. Combining several chains. In this section we extend the notion of a chain function to cover common biological examples in which the regulatee state is a boolean function of several chains. Frequently, a combination of several signals influences the transcription of a single regulatee via several pathways that carry these signals to the nucleus, and a regulation function that combines them together. Here, we formalize this situation by modeling each signal transduction pathway by a chain function, and letting the outputs of these paths enter a boolean gate.

Define a *k-chain function* f as a boolean function which is composed of k chain

functions over disjoint sets of regulators that enter a boolean gate $G(f)$. Let f^i be the ith chain function and let g^i_j denote the jth regulator in f^i. The output of the function is $G(infl(g^1_1), \ldots, infl(g^k_1))$.

In the following we present several biological examples for k-chain functions that arise in transcriptional regulation in different organisms: The lac operon [21] codes for lactose utilization enzymes in *E. coli*. It is under both negative and positive transcriptional control. In the absence of lactose, lac-repressor protein binds to the promoter of the lac operon and inhibits transcription. In the absence of glucose, the level of cAMP in the cell rises, which leads to the activation of CAP, which in turn promotes transcription of the lac operon. In our formalism, the lac operon is controlled by a 2-chain function with an AND gate. The chains are $f^1(g^1_2, g^1_1) = f^1$(lactose, lac-repressor), with control pattern 11, and $f^2(g^2_3, g^2_2, g^2_1) = f^2$(glucose, cAMP, CAP), with control pattern 100. Other examples of 2-chains with AND gates are the regulation of arginine metabolism and galactose utilization in yeast [18]. A 2-chain with an OR gate regulates lysine biosynthesis pathway enzymes in yeast [18].

These examples motivate us to restrict attention to gates that are either OR or AND. We first show that we can distinguish between OR and AND gates. We then show how to reconstruct k-chain functions in the case of OR and later extend our method to handle AND gates.

Denote the output of f^i by O_i. If $O_i = 1$ in wild type, we call f^i a 1-*chain* and, otherwise, a 0-*chain*. A regulator g^i_j is called a 0-*regulator* (1-*regulator*) if its perturbation produces $O_i = 0$ ($O_i = 1$). Let k_0 (k_1) be the number of 0-regulators (1-regulators) in f. A block is called a 0-*block* (1-*block*), if it consists of 0-regulators (1-regulators).

LEMMA 12. *Given a k-chain function f with gate $G(f)$ which is either AND or OR, $k \geq 2$, we can determine, using $O(N^2)$ experiments of order at most 2, whether $G(f)$ is an AND gate or an OR gate.*

Proof. We perform N typing experiments. If $w = 0$ and $\bar{W} = \emptyset$, then $G(f)$ is an AND gate. If $w = 1$ and $\bar{W} = \emptyset$, then $G(f)$ is an OR gate. Otherwise, $\bar{W} \neq \emptyset$. In this situation the cases of $w = 0$ and $w = 1$ are similarly analyzed. We describe only the former.

If $w = 0$, we have to differentiate between the case of an OR gate, whose inputs are all 0-chains, and the case of an AND gate, whose inputs are one 0-chain and $(k-1)$ 1-chains. To this end we perform all comparisons of a W-gene and a $\bar{W}$-gene. Let T be the set of genes g such that the result of a $\{g, g'\}$-perturbation is w for every $g' \in \bar{W}$. Then $T \neq \emptyset$ iff $G(f)$ is an AND gate. □

We now study the reconstruction of an OR gate. Let S be the (possibly empty) set of regulators that reside in one of the first blocks (i.e., the blocks containing g^i_1), that are also 1-blocks. We observe that a perturbation of any regulator in S results in $state(g_0) = 1$ regardless of any other simultaneous perturbations we may perform. Hence, determining the specific chain to which an element from S belongs is not possible. Therefore, our reconstruction will be unique up to the ordering within blocks and up to the assignment of the regulators in S to their chains. The next lemma handles the case $w = 0$. The subsequent lemma treats the case $w = 1$.

LEMMA 13. *Given a k-chain function f with an OR gate and assuming that $w = 0$, we can reconstruct f using N typing experiments and $(N - k_1)k_1$ comparisons.*

Proof. We perform N typing experiments. Then, for each 1-regulator b, we perform all possible comparisons, thereby identifying all 0-regulators that succeed b in its chain. This completes the reconstruction. □

LEMMA 14. *Let f be a k-chain function with an OR gate. Assume that $w = 1$, and let r be the number of 1-chains entering the OR gate. Then f can be reconstructed using $O(N^r + Nn)$ experiments of order at most $r + 2$.*

Proof. First, we determine r, the minimum order of an experiment that will produce output 0 for f. For $i = 1, 2, \ldots$ we perform all possible i-order experiments; r is determined as the smallest i for which we obtain output 0. In total we perform $O(N^r)$ experiments. We call the set of perturbed genes in an r-order experiment which results in output 0, a *reset combination*.

Next, we reconstruct the 1-chains. Fix an arbitrary reset combination R. For every $a \in R$ we perform a set of experiments of order $r + 1$ as follows: For every reset combination $R' \supset R \setminus \{a\}$ with $a \notin R'$, we perturb R' and in addition each other gene, one at a time, recording those that produce output 1 as 1-regulators. For every a, the sets of 1-regulators discovered in these experiments form a linear order under set inclusion. The 1-regulators that are *not* common to all these sets are exactly the 1-regulators (that are not in S) of the chain that includes a. For each 0-regulator in $R' \setminus R$ our experiments determine the 1-regulators that succeed it in this chain. Thus, we can infer all the 1-chains. The total number of experiments performed is $O(Nk_0)$.

Finally, we reconstruct the 0-chains. To this end we perturb the 1-regulators in R, thereby deactivating the 1-chains and reducing the problem of reconstructing the 0-chains to that of reconstructing a $(k - r)$-chain function with an OR gate and $w = 0$ (removing the already discovered regulators of the 1-chains from consideration). This is done by applying the reconstruction method of Lemma 13 using $O(Nk_1)$ experiments of order at most $r + 2$. The assignment of 1-regulators in S will remain uncertain. ∎

Note that for $k = 1$ the above algorithms will reconstruct a single chain. Indeed, for $w = 0$ the algorithm of Lemma 13 coincides with that of section 3, and for $w = 1$, applying the algorithm of Lemma 14 we shall discover that $r = k = 1$. Further note that for every reconstructed chain we can identify whether its first block is a 1-block (i.e., contains genes in S). This is simply done by computing for that chain the value of $state(g_n) \oplus (\oplus_i y_i)$ on its known members and comparing it to the chain's output. Last, note that if k is known and $r = k$, then the order of the experiments that are required to reconstruct the k-chain is at most $r + 1$, since f contains no 0-chains.

The reconstruction method for the case of an OR gate can be used for the reconstruction of an AND gate as well, by exchanging the roles of 0 and 1 in the above description. This gives rise to the following result:

THEOREM 15. *A k-chain function with an OR or an AND gate can be reconstructed using $O(N^k)$ experiments of order at most $k+1$. The reconstruction requires $\Omega(\binom{N}{k}/k)$ experiments of this order.*

Proof. The upper bound follows from Lemmas 12, 13, and 14 and the duality of AND and OR gates. For the lower bound consider a k-chain function with an OR gate consisting of k 1-chains, each of which contains a single 0-regulator. Such a function has a single reset combination, which must be identified in the process of reconstructing the chain. Since each experiment of order $k + 1$ can test at most k combinations, $\Omega(\binom{N}{k}/k)$ experiments are required for the reconstruction. ∎

5. A biological application. The methods we presented above can be applied to reconstruct chain functions from biological data. We describe one such application to the reconstruction of the yeast galactose regulation function, for which some of the required perturbations have been performed. We show that one additional experiment suffices to fully reconstruct the regulation function.

The galactose utilization in the yeast *Saccharomyces cerevisiae* [18] occurs in a biochemical pathway that converts galactose into glucose-6-phosphate. The transporter gene gal2 encodes a protein that transports galactose into the cell. A group of enzymatic genes, gal1, gal7, gal10, gal5, and gal6, encode the proteins responsible for galactose conversion. The regulators gal4p, gal3p, and gal80p control the transporter, the enzymes, and to some extent each other (Xp denotes the protein product of gene X). In the following, we describe the regulatory mechanism. gal4p is a DNA binding factor that activates transcription. In the absence of galactose, gal80p binds gal4p and inhibits its activity. In the presence of galactose in the cell, gal80p binds gal3p. This association releases gal4p, promoting transcription. This mechanism can be viewed as a chain function, where $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal3, gal80, gal4)$, and the corresponding control pattern is 0110 (see also [9]). The gal7, gal10, and gal1 regulatees are also negatively controlled by another chain $f^2(g_2^2, g_1^2) = f^2(glucose, mig1)$ with control pattern 01. The two chains are combined by an AND gate (see Figure 2(A)).

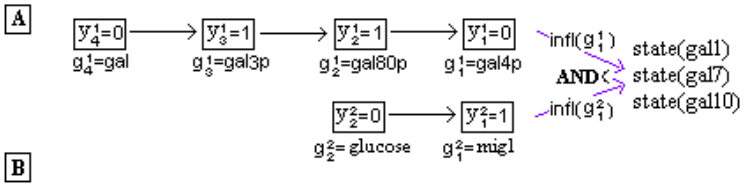
Ideker et al. [14] performed several experiments to interrogate the galactose utilization mechanism. In these experiments glucose was absent from the media. Consequently, the output of f^2 was always 1, and hence we shall focus on the reconstruction of f^1 using the experimental data of [14]. Using the discretization procedure employed by Ideker et al. [14], the measured wild-type levels of gal3, gal80, and gal4 were 1, in accordance with our model assumption. The wild-type level of galactose was also 1.

Assuming we know the group of four regulators, we need, according to Proposition 4, a total of 4 typing experiments and 3 comparisons (since only gal80 is of type W) to reconstruct the chain. Notably, all 4 typings and 2 of the 3 comparisons[2] were performed by Ideker et al. [14] (see Figure 2(B)). Using the same discretization procedure, the experiments yielded the correct results for all three regulatees. The results suggest two possible chain functions: $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal3, gal80, gal4)$ or $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal80, gal3, gal4)$, both with control pattern 0110. The missing experiment is a comparison of gal80 and gal3. A correct result of this experiment will lead to full and unique reconstruction of the chain function.

6. Concluding remarks. In this paper we studied the computational problems arising when wishing to reconstruct regulation relations using a minimum number of experiments, assuming that the experiments' results are noiseless. We restricted attention to common biological relations, called chain functions, and exploited their special structure in the reconstruction. We also suggested an extension of that model, which combines several chain functions, and studied some of the same reconstruction questions for the extended model. On the practical side, we have shown an application of our reconstruction scheme for inferring the regulation of galactose utilization in yeast.

The task of designing optimal experimental settings is fundamental in meeting the great challenge of regulatory network reconstruction. While this task entails coping with complex interacting regulation functions and noisy biological data, we chose here to focus on the reconstruction of a single regulation relation of a single regulatee and assume that the function can be studied in isolation. Hence, upon any perturbation, none of the other regulators change their states. Another major

[2]In fact, the gal80Δgal4Δ-gal experiment was of order 3 but allowed the comparison of gal80 and gal4.



B

Ideker et al. Experiment name	galactose	gal3	gal80	gal4	regulatee	
wt-gal	0	1	1	1	$\bar{w}$	Typing experiments
gal3Δ+gal	1	0	1	1	$\bar{w}$	
gal80Δ+gal	1	1	0	1	w	
gal4Δ+gal	1	1	1	0	$\bar{w}$	
gal80Δ-gal	0	1	0	1	w	Comparison experiments
gal80Δgal4Δ-gal	0	1	0	0	$\bar{w}$	
not perfomed	1	0	0	1	?	

FIG. 2. *Galactose pathway regulation.* (A) *The* 2*-chain function regulating gal*1*, gal*7*, and gal*10 *transcription.* (B) *Typing and comparison experiments performed by Ideker et al.* [14].

assumption is that the wild-type state of all regulators (except possibly g_n) is 1. This assumption, which is necessary for the analysis (e.g., Lemma 3) is commonly held in undelayed biological systems, where all the regulators exist in a certain basal level and the signal can propagate fast (e.g., MAPK systems in unicellular organisms such as yeast and multicellular organisms including humans, reviewed in [17]). Regulations that involve production of absent regulators are typically (slow) temporal processes. Our analysis should be extended in order to deal with such complex regulations and temporal processes.

This analysis focuses on theoretical complexity of regulation reconstruction, assuming perturbation experiments that measure (accurately) only gene states. It is clear, however, that other experimental techniques (e.g., interaction measurements [7, 20]) might help to constrain the reconstruction and reduce the solution space. In a practical approach, diverse data sources should be incorporated, and the experiments should be designed dynamically and take into consideration the experimental noise. The theoretical analysis here could hopefully serve as a component in such a practical experimental design.

REFERENCES

[1] T. AKUTSU, S. KUHARA, O. MARUYAMA, AND S. MIYANO, *Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model*, Theoret. Comput. Sci., 298 (2003), pp. 235–251.

[2] N. ALON, M. BLUM, A. FIAT, S. KANNAN, M. NAOR, AND R. OSTROVSKY, *Matching nuts and bolts*, in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1994, pp. 690–696.

[3] M. ANTHONY, *The Sample Complexity and Computational Complexity of Boolean Function Learning*, Tech. Report LSE-CDAM-2002-13, London School of Economics and Political Science, London, UK, 2002.

[4] N. H. BSHOUTY, *Exact learning Boolean function via the monotone theory*, Inform. and Comput., 123 (1995), pp. 146–153.

[5] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[6] J. Derisi, V. Iyer, and P. Brown, *Exploring the metabolic and genetic control of gene expression on a genomic scale.*, Science, 282 (1997), pp. 699–705.
[7] A. H. Tong et al., *Global mapping of the yeast genetic interaction network*, Science, 303 (2004), pp. 808–13.
[8] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, *Using Bayesian networks to analyze expression data*, J. Comp. Biol., 7 (2000), pp. 601–620.
[9] I. Gat-Viks and R. Shamir, *Chain functions and scoring functions in genetic networks*, Bioinformatics, 19, Supplement 1 (2003), pp. 108–117.
[10] I. Gat-Viks, R. Shamir, R. M. Karp, and R. Sharan, *Reconstructing chain functions in genetic networks*, in Proceedings of the Ninth Pacific Symposium on Biocomputing (PSB'04), 2004.
[11] M. C. Gustin, J. Albertyn, M. Alexander, and K. Davenport, *Map kinase pathways in the yeast Saccharomyces cerevisiae*, Microbiol. Mol. Biol. Rev., 62 (1998), pp. 1264–1300.
[12] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer, *Co-clustering of biological networks and gene expression data*, Bioinformatics, 18, Supplement 1 (2002), pp. 145–154.
[13] S. Hohmann, *Osmotic stress signaling and osmoadaptation in yeasts.*, Microbiol. Mol. Biol. Rev., 66 (2002), pp. 300–372.
[14] T. Ideker et al., *Integrated genomic and proteomic analyses of systematically perturbed metabolic network*, Science, 292 (2001), pp. 929–933.
[15] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel, *Discovering regulatory and signaling circuits in molecular interaction networks.*, Bioinformatics, 18, Supplement 1 (2002), pp. 233–240.
[16] T. Ideker, V. Thorsson, and R. M. Karp, *Discovery of regulatory interaction through perturbation: Inference and experimental design*, in Proceedings of Pacific Symposioum in Biocomputing, 2000, pp. 305–316.
[17] G. L. Johnson and R. Lapadat, *Motigen-activated protein kinase pathways mediated by ERK, JNK, and p*38 *protein kinases*, Science, 298 (2002), pp. 1911–12.
[18] E. W. Jones, J. R. Pringle, and J. R. Broach, eds., *The Molecular and Cellular Biology of the Yeast Saccharomyces: Gene Expression*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1992.
[19] J. Komlós, Y. Ma, and E. Szemerédi, *Matching nuts and bolts in* $o(n \log n)$ *time*, SIAM J. Discrete Math., 11 (1998), pp. 347–372.
[20] T. I. Lee et al., *Transcriptional regulatory networks in Saccharomyces Cerevisiae*, Science, 298 (2002), pp. 799–804.
[21] F. C. Neidhardt, ed., *Escherichia coli and Salmonella: Cellular and Molecular Biology*, ASM Press, 1996.
[22] D. Pe'er, A. Regev, and A. Tanay, *Minreg: Inferring an active regulator set*, Bioinformatics, 18, Supplement 1 (2002), pp. 258–267.
[23] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller, *Rich probabilistic models for gene expression*, Bioinformatics, 17, Supplement 1 (2001), pp. 243–252.
[24] A. Tanay and R. Shamir, *Computational expansion of genetic networks*, Bioinformatics, 17, Supplement 1 (2001), pp. 270–278.

Modeling and Analysis of Heterogeneous Regulation in Biological Networks.

Irit Gat-Viks*† Amos Tanay*† Ron Shamir*

May 16, 2004

Abstract

In this study we propose a novel model for the representation of biological networks and provide algorithms for learning model parameters from experimental data. Our approach is to build an initial model based on extant biological knowledge, and refine it to increase the consistency between model predictions and experimental data. Our model encompasses networks which contain heterogeneous biological entities (mRNA, proteins, metabolites) and aims to capture diverse regulatory circuitry on several levels (metabolism, transcription, translation, post-translation and feedback loops among them).

Algorithmically, the study raises two basic questions: How to use the model for predictions and inference of hidden variables states, and how to extend and rectify model components. We show that these problems are hard in the biologically relevant case where the network contains cycles. We provide a prediction methodology in the presence of cycles and a polynomial time, constant factor approximation for learning the regulation of a single entity. A key feature of our approach is the ability to utilize both high throughput experimental data which measure many model entities in a single experiment, as well as specific experimental measurements of few entities or even a single one. In particular, we use together gene expression, growth phenotypes, and proteomics data.

We tested our strategy on the lysine biosynthesis pathway in yeast. We constructed a model of over 150 variables based on extensive literature survey, and evaluated it with diverse experimental data. We used our learning algorithms to propose novel regulatory hypotheses in several cases where the literature-based model was inconsistent with the experiments. We showed that our approach has better accuracy than extant methods of learning regulation.

*School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. {iritg,amos,rshamir}@post.tau.ac.il.
†These authors contributed equally to this work.

1 Introduction

Biological systems employ heterogeneous regulatory mechanisms that are frequently intertwined. For example, the rates of metabolic reactions are strongly coupled to the concentrations of their catalyzing enzymes, which are themselves subject to complex genetic regulation. Such regulation is in turn frequently affected by metabolite concentrations. Metabolite-mRNA-enzyme-metabolite feedback loops have a central role in many biological systems and exemplify the importance of an integrative approach to the modeling and learning of regulation.

In this work we study steady state behavior of biological systems that are stimulated by changes in the environment (e.g., lack of nutrients) or by internal perturbations (e.g., gene knockouts). Our model of the system contains variables of several types, representing diverse biological factors such as mRNAs, proteins and metabolites. Interactions among biological factors are formalized as regulation functions which may involve several types of variables and have complex combinatorial logic. Our model combines metabolic pathways (cascades of metabolite variables), genetic regulatory circuits (sub-networks of mRNAs and transcription factors protein variables), protein networks (cascades of post-translational interactions among protein variables), and the relations among them (metabolites may regulate transcription, enzymes may regulate metabolic reactions). We show how such models can be built from the literature and develop computational techniques for their analysis and refinement based on a collection of heterogeneous high-throughput experiments. We develop algorithms to learn novel regulation functions in lieu of ones that manifest inconsistency with the experiments.

Most current approaches to the computational analysis of biological regulation focus on transcriptional control. Both discrete (e.g., [3]) and probabilistic methods (e.g., [9]) use gene expression data and attempt to learn a regulatory structure among genes and to create a predictive model that fits the data. The computational models used in these studies involve numerous simplifying assumptions on the nature of genetic regulation. Among the more problematic of these simplifications are a) the use of mRNA levels to model the activity of transcription factor proteins, b) the lack of consideration for the state of the medium in which the experiment was done and c) the assumption of acyclic regulation structure that prevents the adequate modeling of feedback loops. As a consequence of these limitations, simple genetic networks tools are rarely used in practical biological settings. A more fruitful approach for learning regulation involves the coarser notion of regulatory modules, with [14] or without [1, 17] explicit learning of regulatory functions that define them. Module-based methods are relatively robust to noise and in some cases can tolerate the gross simplification described above. However, models generated by these methods are coarse and limited in their level of detail.

Our study aims to overcome some of the limitations of prior art by taking an approach that is innovative in combining several key aspects:

• We model a variety of variables types, extending beyond gene network studies, that focus on mRNA, and metabolic pathways methods, that focus on metabolites. Consequently, our model can express the environmental conditions and the effects of translation regulation and post translational modifications.

• Our approach allows handling feedback loops as part of the inference and learning process. This is crucial for adequate joint modeling of metabolic reactions and genetic regulation.

• We build an initial model based on prior knowledge, and then aim to improve (expand) this model based on experimental data. A similar approach was employed in [16] for transcription regulation only. We show that formal modeling of the prior knowledge allows the interpretation of high throughput experiments on a new level of detail.

• Our algorithms learn new transcription regulation functions by analyzing together gene expression, protein expression and growth phenotypes data.

Our methodologies and ideas were implemented in a new software tool called MetaReg. It facilitates evaluation of a model versus diverse experimental data, detection of variables that manifest inconsistencies between the model and the data, and learning optimized regulation functions for such variables. We used MetaReg to study the pathway of lysine biosynthesis in yeast. We performed an extensive literature survey and organized the knowledge on the pathway into a model consisting of about 150 variables. In the process of model construction, we reviewed the results of many low throughput experiments and included in the model the most plausible regulation function of each variable. We assessed the model versus a heterogeneous collection of experimental results, consisting of gene expression, protein expression and phenotype growth sensitivity profiles. In general, the model agreed well with the observations, confirming the effectiveness of our strategy. In several important cases, however, inconsistencies between measurements and model predictions indicated gaps in the current biological understanding of the system. Using our learning algorithm we generated novel regulation hypotheses that explain some of these gaps. We also showed that our method attains improved accuracy in comparison to extant network learning methods.

The paper is organized as follows. In Section 2 we introduce the model and define some notation. In Section 3 we show how to take feedback loops into account and how to use the model to infer the system state given an environmental stimulation. In Section 4 we introduce our mathematical formulation of experimental data and model scoring scheme and in Section 5 we develop optimization algorithms for the learning of regulation functions. Section 6 presents our results on the lysine pathway and its regulation. Some proofs and experimental details appear in an appendix.

2 The model

We first define a formal model for biological networks. A *model* M is a set U of *variables*, a set $S = \{1, \ldots, k\}$ of discrete *states* that the variables may attain, and a set of *regulation functions* $f_v : S^{|N(v)|} \rightarrow S$ for each $v \in U$. f_v defines the state of a regulated variable v (called a *regulatee*) as a function of the states of its *regulator* variables $N(v) = \{r_v^1, \ldots, r_v^{d_v}\}$. We define the set of *stimulators* U_I to include all variables with zero indegree. The *model graph* of M is the digraph $G_M = (U, A)$ representing the direct dependencies among variables, i.e., $(u, v) \in A$ iff $u \in N(v)$. For convenience we assume throughout that regulation functions can be computed in constant time.

A *model state* s is an assignment of states to each of the variables in the model, $s : U \rightarrow S$. A model *stimulation* is an assignment of states to all the model stimulators, $q : U_I \rightarrow S$.

In this paper we shall use the model logic primarily for the determination of modes. For a model M and state s, we say that s *agrees with* M *on* v if $f_v(s(r_v^1), \ldots, s(r_v^{d_v})) = s(v)$. We call a model state s of M a *mode* if s agrees with M on every $v \in U \setminus U_I$. A mode is thus a steady state of the system. States representing non-steady state behavior of the system, which may be adequate for the representation of temporal processes, are outside the scope in this work. Since our biological models represent a combination of diverse regulation mechanisms, operating in different time scales (metabolic reactions are orders of magnitude faster than transcription regulation), a realistic temporal model is a considerable challenge that should be carefully dealt with in future work. The steady state assumption is in wide use (e.g., [3, 9]) and was proved flexible enough in our empirical studies. Figure 1 illustrates a simple model and its modes.

We now describe the biological semantics of a model. V includes four *types* of variables: (a) mRNAs (b) active proteins that serve as enzymes or regulators (c) internal metabolites, which represent the metabolite derivatives in the pathway under study (d) external metabolites, which represent different environmental conditions and specify the nutritional concentrations in the medium. The external metabolites are assumed to be determined by the experimenter, and their level is unaffected

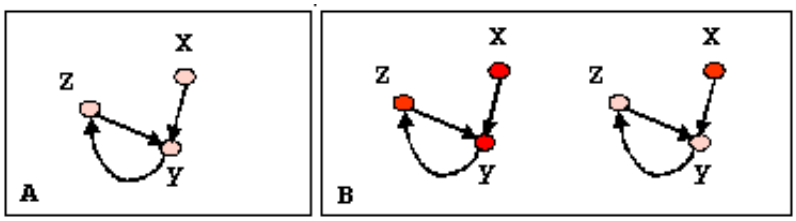


Figure 1: **A simple model**. The model includes one stimulator X, regulating a positive feedback loop of two variables Y and Z. We assume a binary state space (on-dark, off-light). f_z is the identity function and $f_y = s(x)\ AND\ s(z)$. When the stimulator state is off (A), a unique mode exists. If the stimulator state is on (B), two different modes are possible, one in which the cycle is on and the other in which the cycle is off.

by other variables in the model, so they will serve as part of our stimulator set. The levels of the mRNAs, proteins and internal metabolites are controlled by other variables via regulation functions that manifest transcriptional, translational, post-translational and metabolic control mechanisms. The stimulators determine the "boundary condition" of the model. For example, in lysine metabolism, the level of the internal lysine metabolite is influenced by lysine transport into the cell, by the yield of the lysine biosynthetic pathway, by the rate of lysine degradation, and by the rate of lysine utilization in proteins biosynthesis. The external lysine level, on the other hand, is assumed to be determined and kept fixed by the experimenter throughout the experiment.

3 Computing modes

Given a model stimulation q we would like to compute the set of model's modes whose stimulators states coincide with those of q. This will be the first step in using a model to infer the state of the system under a certain condition.

A *q-mode* of a model M and stimulation q is a mode m such that for each $v \in U_I$, $q(v) = m(v)$. We denote the set of q-modes by $Q_{q,M}$. A model M with acyclic graph G_M is called a *simple model*. We note that q-modes are unique and easily computable for simple models: Given a stimulation q and a topological ordering on the graph's nodes (which exists, since the graph is acyclic), we can compute the q-mode by calculating the state of each variable given its regulators' states. In summary:

Claim 1 *Let M be a simple model where $G_M = (U, A)$. For any stimulation q, there is a unique q-mode that can be computed in time $O(|U| + |A|)$.*

In practice, model graphs are not acyclic and feedback loops play a central role in system functionality. In cyclic models, a stimulation q may have no q-modes (in case no steady state is induced by the stimulation), a unique q-mode, or several q-modes. In order to compute the set of q-modes we will first transform a cyclic model into a simple one. Recall that a *feedback set* in a directed graph is a set of nodes whose removal renders the graph acyclic [6]. A feedback set of a model M is a feedback set for the graph G_M. Given a feedback set F, the *auxiliary model* M_F is obtained by changing the regulation functions of the variables in F to null. The graph G_{M_F} is updated accordingly and becomes acyclic, so M_F is simple. Given a set $F' \subseteq F$, we say that a mode m' of M_F is (M, F')-*compatible* if m' agrees with M on every $v \in F'$. In particular, a mode of M_F which is (M, F)-compatible is also a mode for M, since the steady state requirements hold for every $v \in U \setminus F$ (by definition of M_F modes) and for all $v \in F$ (due to the compatibility). Given a mode for M_F, it is easy to check

if it is (M, F')-compatible by calculating f_v for each $v \in F'$. The following algorithm calculates the q-modes of M by using a feedback set F and a topological ordering of G_{M_F}:

Mode Computation Algorithm

- Generate each possible state assignment to F. For the assignment $s_F : F \to S$ do the following:
  - Generate a stimulation q' for M_F by joining q and s_F.
  - Use the topological ordering to compute a (unique) q'-mode m'.
  - If m' is (M, F)-compatible, add it to $Q_{q,M}$.

Hence, we have shown:

Proposition 2 *Given a model M, a feedback set F, and a stimulation q, the q-modes can be computed in $O(k^{|F|}(|U| + |A|))$ time.*

We note that the minimum feedback set problem is NP-hard [12], but approximation algorithms are available [15]. The complexity of our algorithm is exponential in the size of the feedback set, but this is tolerable for the current models we have analyzed. Much larger systems may require heuristics that avoid the exhaustive enumeration of feedback set states we are currently using.

4 Experimental conditions and their inferred modes

An ultimate test for a model is its ability to predict correctly the outcome of biological experiments. We formally represent the data of such experiments as *conditions*. A condition e is a triplet (e_q, e_p, e_s). e_q is a model stimulation defining the environment in which the experiment was performed. e_p is a partial assignment of states to variables in $U \setminus U_I$, and is called a *perturbation*. A perturbation defines a set of variables whose regulation was kept as a particular constant during the experiment. For example, knockout experiments fix the state of mRNAs to zero. e_s is a set of measurements of the states of some variables, and is called an *observed partial state*. We define $e_s(v) = -1$ for variables that were not measured in the experiment. Low throughput experiments (like northern blot or ELISA) typically measure one or few variables in a given condition. High throughput experiments (e.g., gene expression arrays or protein expression profiles) may measure the states of all variables of a particular type. A different type of high throughput experiments are growth sensitivity mutant arrays [4]. Each such array corresponds to many conditions, all with the same stimulation (representing the environment of the experiment), but with different perturbations (different knocked-out genes), and only a single measured variable: the growth level. We will assume that this level corresponds to the yield of the metabolic pathway under study.

Given a condition e we wish to use a model M to compare the possible modes induced by the stimulation e_q with the observed partial state. If the condition involves a perturbation, we first have to update our model accordingly. For simplicity assume this is not the case. We then apply the algorithm from the previous section and compute the set of all e_q-modes. In case more than one exists, we expect the correct one to be most similar to the observed partial state. To assess this similarity we introduce a score function that equals the sum of squared differences between the observed partial state e_s and a e_q-mode. Precisely, given a condition e and an e_q-mode s, we define the discrepancy $D(s, e)$ as $\sum_{v \in U, e_s(v) \neq -1} (s(v) - e_s(v))^2$. The mode with smallest discrepancy will be considered as our *inferred mode*. Its score is called the *model discrepancy* on condition e, i.e., $D(M, e) = \min_{s \in Q_{e_q,M}} D(s, e)$. If no e_q-mode exists, $D(M, e)$ is set to a large constant K. Note that models with loosely defined regulation functions may have a large number of modes per stimulation and consequently suffer from over-fitting of the inference.

5 Learning regulation functions

Given a model and experimental conditions, we wish to optimize one particular regulation function in the model and in this way derive an improved model with lower discrepancy. In this section we discuss the resulting function optimization problem, and show that this problem is NP-hard. We translate the function optimization problem to a combinatorial problem on matrices, and provide a polynomial-time greedy algorithm for it. Finally, we show that the greedy algorithm guarantees a 1/2-approximation for a maximization variant of the function optimization problem.

We focus on one model variable v and fix the set of v's regulators $\{r_v^1, ...r_v^{d_v}\}$. Let $E = \{e^i\}$ be the set of experimental conditions. In order to simplify the presentation, we assume throughout this section that experimental conditions have empty perturbation sets. Given a function $g : S^{d_v} \to S$ we define $M(g, v)$ to be the model M with the single change that $f_v = g$. The *discrepancy score of* g is defined as $\sum_i D(M(g, v), e^i)$.

Problem 1 The function optimization problem. *The problem is defined with respect to a model M, a set of conditions E and a variable $v \in U$. The goal is to find a regulation function $f_v = g$ with an optimal discrepancy score. In other words, we wish to compute $argmin_g \sum_i D(M(g, v), e^i)$.*

In most extant gene networks models [9, 3, 16], an optimal regulation function can be easily learned given the topology of the network. This is done using the multiplicities (or probabilities) of different combinations of observed states for the regulators and regulatee. The main difficulty with our version of the learning problem is that the states of regulators are frequently not observed, and have to be inferred together with the regulation function. A naive algorithm can test all $k^{k^{d_v}}$ functions for the best discrepancy, but this strategy is impractical even for modest k and d_v ($3^{3^3} > 10^{12}$). In fact, the optimization problem is NP-hard (for a proof, see the appendix).

Proposition 3 *The function optimization problem is NP hard.*

We shall translate the function optimization problem to a combinatorial problem on matrices and develop an approximation algorithm to solve it. First, we define an auxiliary matrix and show how to construct it. We define $Q^v_{q,M}$ as the set of model states s which satisfy for all $u \in U_I$, $s(u) = q(u)$ and agree with M on all $u \in U \backslash U_I, u \neq v$. Note that $Q^v_{q,M}$ is a superset of the set of q-modes $Q_{q,M}$ in which we relax the requirement for agreement on v. Given an instance of the learning problem, we form a matrix W^v with a column for each condition and a row for each assignment of states to v and its regulators. We define the matrix entry $w^v_{i,((x_1,...,x_{d_v}),x)}$ as $\min\{D(s, e^i_s) | s \in Q^v_{e^i_q,M}, s(\overline{r}) = \overline{x}, s(v) = x\}$ or K if $Q^v_{e^i_q,M} = \emptyset$, where K is a large constant, $\overline{r} = (r_v^1, \ldots r_v^{d_v})$, $\overline{x} = (x_1, \ldots, x_{d_v})$. In the following algorithm, we show how to compute W^v by relaxing the requirement for v compatibility in the mode computation algorithm. Later we shall show how to use W^v to compute the discrepancy score.

Matrix Construction Algorithm

- Initialize all entries in W^v to K.
- Form a feedback set F such that $v \in F$.
- For each condition i and for each assignment s_F of states of the feedback set do:
  - generate a stimulation q' for M_F by joining e^i_q and s_F.
  - use a topological ordering on G_{M_F} to compute a (unique) q'-mode m' for M_F.
  - If m' is $(M, F \setminus v)$-compatible, compute its discrepancy x.
  - Replace the entry $w^v_{i,((m'(r_v^1),...,m'(r_v^{d_v})),m'(v))}$ by x if the latter is smaller.

Lemma 4 *Given a model M, a set of conditions E and a feedback set F such that $v \in F$, the Matrix Construction Algorithm correctly computes the matrix W^v in $O(k^{d_v+1}|E| + k^{|F|}(|U| + |A|)|E|)$.*

Proof: Matrix entries are computed by minimization of discrepancies over all $(M, F \setminus v)$-compatible modes that have a given regulator/regulatee states. But $(M, F \setminus v)$-compatible modes are exactly the modes in $Q^v_{e^i_q, M}$ which are used in W^v's definition. Therefore, the algorithm correctly computes W^v. The algorithm spends $O(k^{d_v+1}|E|)$ (the size of W^v) time in initialization and $O(k^{|F|}(|U| + |A|)|E|)$ time to compute all mode discrepancies. ■

Lemma 5 *The discrepancy score of a regulation function g equals $\sum_{i=1}^{|E|} \min_{\overline{x} \in S^{d_v}} w^v_{i,(\overline{x}, g(\overline{x}))}$.*

Proof: We will show that for each i, $min_{\overline{x} \in S^{d_v}} w^v_{i,(\overline{x}, g(\overline{x}))} = D(M(g, v), e^i)$.
$D(M(g, v), e^i) = \min_{s \in Q_{e^i_q, M(g,v)}} [D(s, e^i_s)] = \min_{\overline{x} \in S^{d_v}} \min_{s \in Q_{e^i_q, M(g,v)}, s(\overline{r}) = \overline{x}} [D(s, e^i_s)] =$
$\min_{\overline{x} \in S^{d_v}} \min_{s \in Q^v_{e^i_q, M(g,v)}, s(\overline{r}) = \overline{x}, s(v) = g(\overline{x})} [D(s, e^i_s)] = min_{\overline{x} \in S^{d_v}} w^v_{i,(\overline{x}, g(\overline{x}))}$. ■

By the last lemma, the scores of all possible regulation functions can be derived from the matrix W^v. To find the optimal function we first translate the problem to the following combinatorial problem:

Problem 2 The Rows Subset Cover Problem. *We are given a non-negative integer valued $n \times m$ matrix W and a partition of the rows to disjoint subsets $B_1, \ldots, B_l$. A* row subset *R is a set of rows $b^R_1 \in B_1, b^R_2 \in B_2, \ldots, b^R_l \in B_l$. Our goal is to find a row subset with maximal score $c(R) = \sum_{j=1}^{m} \max_{i=1}^{l} w_{b^R_i, j}$.*

In our settings, rows are pairs $(\overline{x}, x)$ and columns are conditions. The subsets B_j are the sets of columns with identical regulator states $\overline{x}$. To formulate the function optimization problem as a row subset cover problem we rewrite $w_{ij} = K - w^v_{ij}$. A selection of $b_i = (\overline{x}, x)$ corresponds to the setting of $f_v(\overline{x}) = x$.

The previous discussion implies that for constant value of d_v and k, the row subset cover problem is NP-hard. A *Greedy Row Subset Algorithm* applies naturally to this problem: We start with an arbitrary row subset S, and repeatedly substitute a row to improve the score, i.e., setting $S \leftarrow (S \setminus \{b^S_i\}) \cup \{b'_i\}$ where $b'_i \in B_i$ and the new S has improved score. The algorithm terminates in a local optimum when no single row substitution can improve the score. Since the score increases at each iteration and all scores are integers bounded by K, the greedy algorithm will terminate after $O(nmK)$ steps. For the function optimization problem, $O(|E||U|k^2)$ is an upper bound on the maximal score and hence on the number of steps. Each step costs $O(|E|k^{d_v+1})$ in order to find an improving substitution, and thus the total cost is $O(|E|^2|U|k^{d_v+3})$.

Proposition 6 *The greedy algorithm guarantees a 1/2-approximation for the Row Subset Cover Problem.*

Proof: Given a row subset R, the score $c(R)$ can be expressed as a sum of terms of the form $c_j(R) = max^l_{i=1}[w_{b^R_i, j}]$. We partition the columns according to $argmax_{i=1,\ldots,l}[w_{b^R_i, j}]$ by defining $P^R_i = \{j | w_{b^R_i, j} \geq w_{b^R_k, j},\ k \neq i\}$ and transforming $P^R_1, ..., P^R_l$ into a partition by arbitrarily breaking ties. We now have $c(R) = \sum_{i=1}^{l} \sum_{j \in P^R_i} [c_j(R)]$.

Let A be an optimal row subset, and let D be the output of the greedy algorithm. To prove the approximation ratio, we will show that $c(A) - c(D) \leq c(D)$. We first rewrite this inequality using

two different column partitions, $\sum_{i=1}^{l}\sum_{j\in P_i^A}[c_j(A)-c_j(D)] \leq \sum_{i=1}^{l}\sum_{j\in P_i^D}[c_j(D)]$. In fact, we will show that the inequality holds separately for each term, i.e., $\sum_{j\in P_i^A}[c_j(A)-c_j(D)] \leq \sum_{j\in P_i^D}[c_j(D)]$.

If $b_i^A = b_i^D$, for each $j \in P_i^A$, $c_j(A) \leq c_j(D)$ since the maximal value for A in column j is $w_{b_i^A,j}$, and solution D can use the same value or a better one. Therefore, $\sum_{j\in P_i^A}[c_j(A)-c_j(D)] \leq 0$ and the inequality holds.

Assume now that $b_i^A \neq b_i^D$. We define a new row subset E_i, which is the same as D except for replacing b_i^D with b_i^A. The replacement decreases the score in certain columns (denoted L^-) and increases it in others (denoted L^+). Call the decrease in the set L^- the *loss* and call the increase in L^+ the *gain* caused by the replacement. The key to the proof will be the fact that the loss must be equal or larger than the gain, or else the greedy algorithm would have not stopped at D. Since $L^- \subset P_i^D$ we have

$$\sum_{j\in P_i^D}[c_j(D)] \geq \sum_{j\in L^-}[c_j(D)-c_j(E_i)] = loss. \tag{1}$$

Next look at L^+ and P_i^A. For $j \in L^+ \setminus P_i^A$ we have $c_j(E_i)-c_j(D) > 0$. For $j \in L^+ \cap P_i^A$ we have $c_j(E_i)-c_j(D) = c_j(A)-c_j(D)$. For $j \in P_i^A \setminus L^+$ we have $c_j(A)-c_j(D) < 0$. Overall we get:

$$gain = \sum_{j\in L^+}[c_j(E_i)-c_j(D)] \geq \sum_{j\in P_i^A}[c_j(A)-c_j(D)]. \tag{2}$$

In summary, $\sum_{j\in P_i^D}[c_j(D)] \geq \sum_{j\in L^-}[c_j(D)-c_j(E_i)] \geq \sum_{j\in L^+}[c_j(E_i)-c_j(D)] \geq \sum_{j\in P_i^A}[c_j(A)-c_j(D)]$. The first and third inequalities follow by (1) and (2), respectively. The second inequality is the observation that the loss exceeds the gain. ■

In practice, we find regulation functions by executing the matrix construction algorithm and applying the greedy algorithm to the obtained matrix. Note that our approximation holds only for the maximization problem. Developing any constant ratio approximation for the minimization problem in its current form is NP-hard. This follows from Proposition 3, since the matrix corresponding to the model in that proof has a minimum score of zero, and thus the decision problem with score zero is NP-hard.

We note that in order to take condition perturbations into account, we have to consider a slightly different model in each condition. For example, if a condition was measured in a strain knocked-out for a specific gene v, we will form a modified model with altered (constant) f_v function and compute its modes and discrepancy as described above. The other algorithms (matrix generation and row selection) remain unchanged.

6 Results

We applied the *MetaReg* modeling scheme and algorithms to study lysine biosynthesis in the yeast *S. cerevisiae*. This system was selected since a) it is a relatively simple metabolic pathway, b) its regulatory mechanisms are relatively well understood, and c) several high throughput datasets which include experimental information pertinent to lysine biosynthesis are available.

6.1 A Model for Lysine Biosynthesis

We have performed an extensive literature survey and constructed a detailed model for lysine biosynthesis and related regulatory mechanisms. Lysine, an essential amino acid, is synthesized in *S. cerevisiae* from α-ketoglutarate via homocytrate and α-aminoadipate semialdehyde (αAASA) in a

linear pathway in which eight catalyzing enzymes are involved. The production of lysine is controlled by several known mechanisms:

(1) Control of enzymes transcription via the general regulatory pathway of amino acids biosynthesis. Starvation for amino acids, purines and glucose, induce the synthesis of GCN4ap [1] which is a transcriptional activator of enzymes catalyzing amino acids biosynthesis in several pathways, including lysine. GCN4ap is controlled on the translation level by the translation initiation machinery. Specifically, GCN2ap (a translation initiation factor 2α kinase) is known to mediate the de-repression of GCN4m translation in nutrient-starved cells. The activity of GCN2ap is induced by high levels of uncharged tRNA under starvation conditions [5].

(2) Transcription control of several catalyzing enzymes is regulated by αAASA. The control is mediated by the LYS14ap transcriptional activator in the presence of αAASA, an intermediate of the pathway acting as a coinducer. αAASA serves as a sensor of lysine production [13].

(3) Feedback inhibition of homocytrate synthase isoenzymes (LYS20ap and LYS21ap) by lysine. The first step of the lysine biosynthetic pathway is catalyzed by LYS20ap and LYS21ap. At high levels of lysine, LYS20ap and LYS21ap are inhibited, and thus the production of the pathway intermediates and of lysine itself is reduced [8].

(4) MKS1ap down-regulates CIT2m expression and hence cytrate-synthase production which is needed for the synthesis of α-ketoglutarate. The resulting limitation of α-ketoglutarate decreases the rate of lysine synthesis. MKS1ap is activated in nutrient-starved cells [7, 18].

In Figure 2, we present the model graph of lysine biosynthesis as described above. The graph includes the lysine biosynthetic pathway, the catalyzing enzymes and their transcription control, and the translation initiation machinery controlling GCN4ap state. The model includes also external amino acids and ammonium (NH3). These are needed as stimulators to represent the environmental conditions enforced on the system. The transport of amino acids and ammonium into the cell is facilitated via specific permeases, and the level of internal amino acids and ammonium is determined by the extracellular metabolites and by the activity of these permeases. The state of internal lysine depends on the lysine transport into the cell and on the yield of the lysine biosynthetic pathway. Note that in order to study the model in relative isolation from other pathways and regulatory systems, we had to exclude some of the known relations (e.g., CIT2 and the Kreb cycle in α-ketoglutarate production, tRNAs in GCN2ap activation). The model graph contains several cycles that correspond to three distinct feedback cycles: general nitrogen control regulation (e.g. GCN2ap $\rightarrow$ GCN4ap $\rightarrow$ LYS1,9m $\rightarrow$ LYS1,9ap $\rightarrow$ ILys $\rightarrow$ GCN2ap), lysin negative regulation (LYS20ap/LYS21ap $\rightarrow$ IHomoCytrate $\rightarrow$ αAASA $\rightarrow$ ILys $\rightarrow$ LYS20ap/LYS21ap) and αAASA positive regulation (e.g. LYS14ap $\rightarrow$ LYS2m $\rightarrow$ LYS2ap $\rightarrow$ αAASA $\rightarrow$ LYS14ap). We used a feedback set F consisting of $GCN2ap$ and IαAASA in all the computations reported below. The complete and annotated list of regulation functions that are part of the model, is available upon request.

We used the state space $S = \{0, 1, 2\}$. In our experiments, the definition of compatibility used for the calculation of q-modes was relaxed a bit to include also cases where $m'(v)$ and $f_v(m'(r_v^1), \ldots, m'(r_v^{d_v}))$ are both non-zeros (i.e., cases where inferred state was 1 and observation 2 or vice versa are not considered violation of compatibility). In other words, $D(i, j)$ was $(i - j)^2$ for all states $\{i, j\} \neq \{1, 2\}$, but $D(1, 2)$ and $D(2, 1)$ were set to 0. This was done to allow more flexibility in the model and to focus more on major discrepancies.

[1] We use variable affixes to indicate types. m suffix: mRNA, ap suffix: active protein. Metabolites names are prefixed to indicate their type, I: internal, E: external.

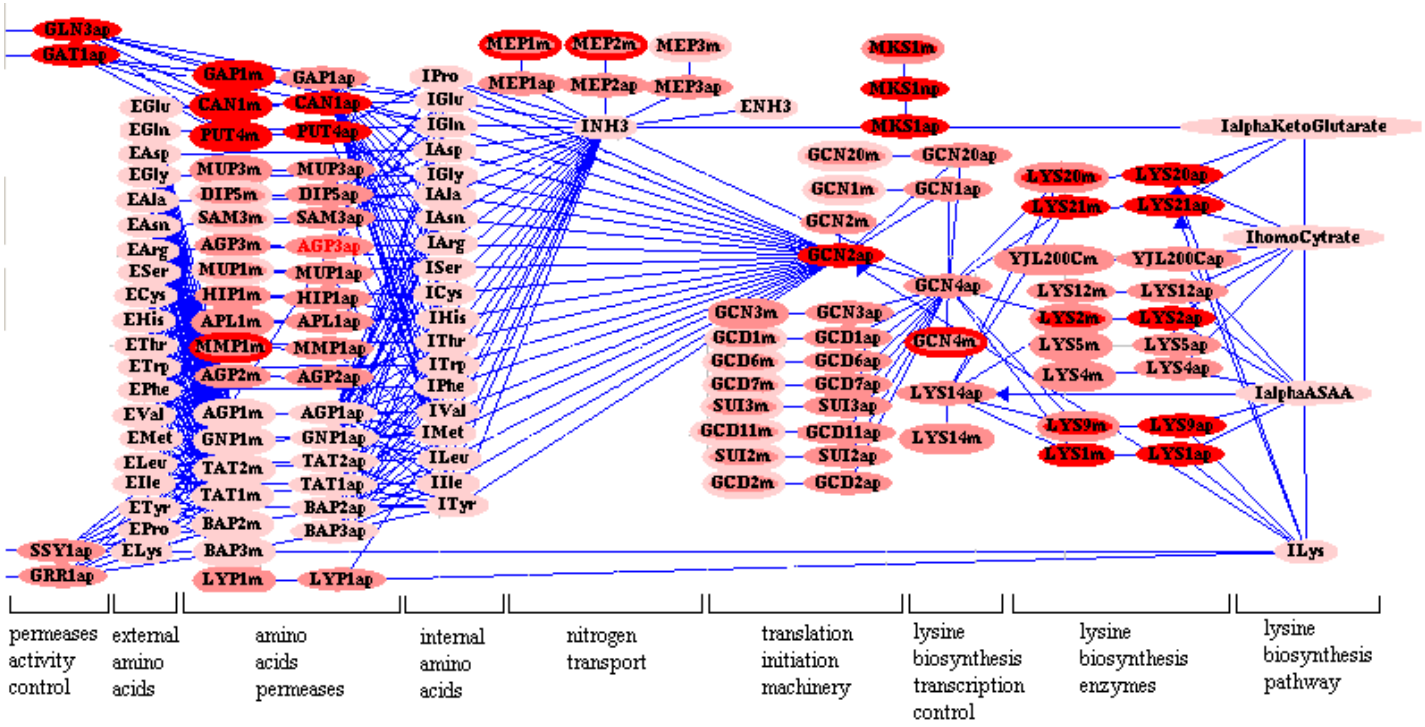


Figure 2: **The model graph of lysine biosynthesis in S. cerevisiae.** Variables are represented by nodes. Arcs lead from each regulator to its regulatees. All arc directions are at any angle to the right or straight down, unless otherwise indicated. The model includes also a regulation function for each regulated variable. These functions are not shown here. Node colors indicate the mode inferred states and the observed states in condition of nitrogen depletion after 2 days. Internal node color: inferred state. Node boundaries: observed state. Red(dark): state= 2. Dark pink(grey): 1. Light pink(light grey): 0. The representation enables us to view the disagreements as color contrasts between the observed and inferred states. For example, LYS9m (bottom right) inferred state is 2 while its observed state is 1.

6.2 Data Preparation

We formed a heterogeneous dataset from five different high-throughput experiments: (a) 10 expression profiles in nitrogen depletion medium after 0.5h, 1h, 2h, 4h, 8h, 12h, 1d, 2d, 3d, 5d of incubation [10]. (b) 5 expression profiles in amino acid starvation after 0.5h, 1h, 2h, 4h, 6h of incubation [10]. (c) 10 microarray experiments of His and Leu starvations and various GCN4 perturbations [5]. (d) protein and mRNA profiles of wild type strain in YPD and minimal media [19]. (e) 80 Growth sensitivity phenotypes [4]. The growth phenotypes were measured for each of a collection of ten gene-deletion mutant strains in eight conditions: Lys, Trp and Thr starvation, three minimal media and two YPG conditions.

To incorporate these data into our framework, we generated conditions from each of the experiments. To this end, we identified the stimulation and perturbation that match each experiment from the respective publication. We then converted the data into a set of observed states. In the Appendix, we define the conversion process of each data set. Note that some of the experiments translate directly to observed states (e.g., growth profiles) and some must be manipulated further (e.g., mRNA ratios of two conditions).

6.3 Model discrepancy

For each of the high throughput conditions in (a) through (d) we computed inferred modes and compared them to the observed states. Recall that the environment defined by the condition's stimulation gives rise to a set of possible inferred modes, and we choose the inferred mode which fits the observed states best. Typically, there are only few modes per condition in the lysine model, confirming the relatively good characterization of the system by the model.

Figure 3A summarizes the comparison between inferred modes and observed states for expression conditions. Figure 3B does the same for growth sensitivity data. In general, there is good agreement between the inferred and observed states. The matrix view highlights conditions and variables in which the observations deviate from the model predictions.

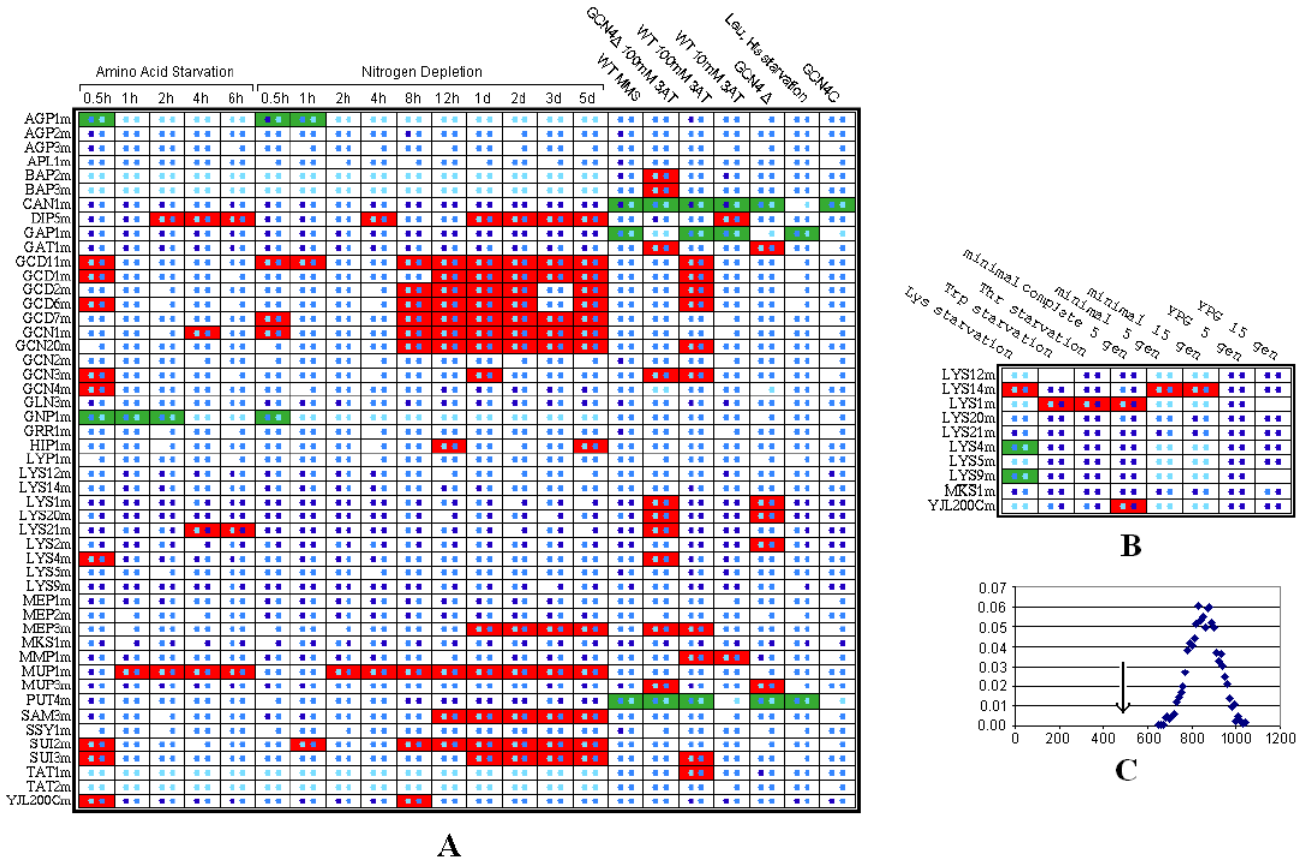


Figure 3: **Model Discrepancy (A) Discrepancy matrix for the expression data.** Columns correspond to conditions and rows correspond to mRNA variables. Each cell contains two small squares: observed (left) and inferred (right) states of the row variable in the column condition. State colors: Cyan (light gray):0, light blue (gray):1, dark blue (black):2. The background color of the cells emphasizes critical disagreement, where the inferred state is zero and the observed state is not (green or light gray), or vice versa (red or gray). **(B) Discrepancy matrix for the phenotype data.** Each cell represents a condition, which is a combination of certain environmental nutrients and one gene deletion. Columns correspond to the nutritional environment (i.e., the medium), and rows correspond to the knocked-out variable. Each cell contain two small squares: observed (left) and inferred (right) state of the internal lysine metabolite (the ILys variable) in this condition. Colors are as in (A). **(C). Distribution of model discrepancy scores for randomly shuffled data sets.** X axis: total model discrepancy. We generated the distribution by computing model discrepancy for 50 random data sets. The discrepancy of the real data set is 494 (arrow), much lower than the minimal discrepancy measured in the shuffled sets.

Before analyzing the deviations, we verified the specificity of the total discrepancy. Since the mode computation algorithm involves selection of one mode from several possibilities in each condition, we wanted to verify that this process does not cause over-fitting. To this end, we generated randomly shuffled data sets in which we swapped the states between variables of the same type. Figure 3C shows the discrepancy distribution obtained from this experiment, and supports the high specificity of the lysine model discrepancy.

We next examined the biological implication of two major deviations of the inference from the experimental data: First, the transcription of the translation initiation machinery (GCD1,2,6,7,11, GCN1,20, SUI2,3) is repressed in the later phases (8h-5d) of the nitrogen depletion experiment, and this effect is not predicted by the model. Moreover, the transcription of the ammonium permeases MEP1 and MEP2 is consistently activated in nitrogen depletion. To the best of our knowledge, the explanation for these observations is still unclear. However, there is some evidence for involvement of the TOR signaling pathway in the regulation of this response [2]. Second, the transcription of the lysine biosynthesis catalyzing enzymes is known to be activated by both LYS14ap and GCN4ap, but the exact combinatorial regulation function is unknown. Since they are both known to be activators, we originally modeled the regulation function of the catalyzing enzymes (LYS1,2,9,20,21) simply as the sum of LYS14ap and GCN4ap. In most catalyzing enzymes, there is a clear inference deviation in two conditions with GCN4Δ strain (Figure 3A, 3rd and 6th columns from right). In addition, the growth phenotypes of LYS14 deletion strain (Figure 3B, second row) deviate from their inferred states in all conditions with nutritional limitation of lysine. Therefore, the regulation function we originally modeled for the lysine biosynthesis catalyzing enzymes is apparently not optimal.

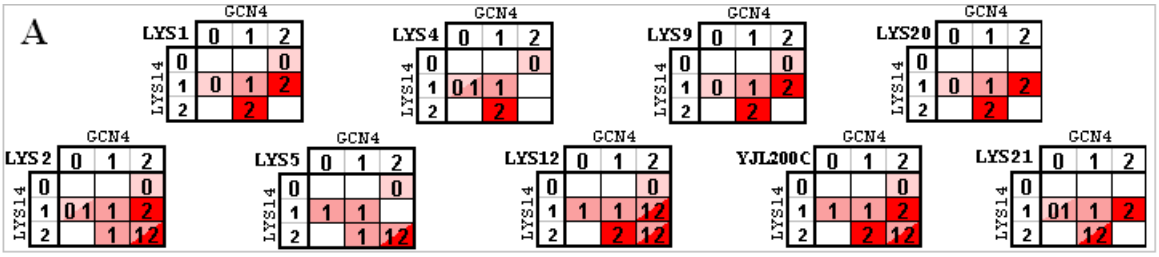
6.4 Learning improved regulation functions

To refine our understanding of the combinatorial regulation scheme involving LYS14ap and GCN4ap we applied our learning algorithm to the regulation functions of LYS1,2,4,5,9,12,20,21. For each one, we computed the discrepancy matrix and selected an optimal regulation function using the learning algorithm outlined in Section 5. To estimate the confidence of our learned functions we used a bootstrap procedure as follows. We generated 1000 datasets each containing a random subset of 80% of the original set of conditions. For each random dataset we recalculated the optimal regulation functions for each of the enzymes. The *confidence* of the function entry $f_v(x_1, \ldots, x_{d_v}) = y$ was defined as the fraction of times y was learned as the function value for the regulators values $x_1, \ldots, x_{d_v}$. In case of ties (several function outcomes with equal scores), we split the count among the candidate outcomes. Results are summarized in Figure 4A,B,C.

Based on the optimal functions, we identify two enzyme sets that share a regulatory program. The expression of genes in the first set (LYS1,9,20 and possibly LYS4 and LYS21) is dependent on the presence of both LYS14 and GCN4. Both transcription factors seems to drive the transcription of enzymes in this set linearly. The second set, including LYS5, LYS12 and YJL200C require LYS14 but not GCN4 for basal expression levels. For LYS5 it seems that GCN4 may not be a regulator at all, possibly since LYS5 is not a catalyzing enzyme in the pathway under study. We note that the combination of expression and growth phenotype information was crucial for deriving this conclusion. For example, when using expression data alone, the rows with LYS14p=0 are completely undefined.

6.5 Cross validation

We tested the predictive quality of *MetaReg* by performing leave-one-out cross validation. For the test, we used the set of enzymes $L = \{$LYS1,2,4,5,9,12,20,21m$\}$ as regulatees and GCN4ap, LYS14ap,

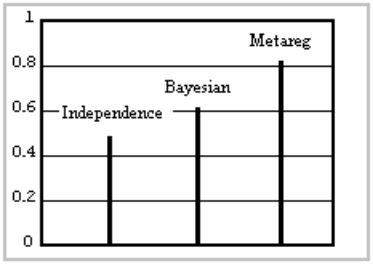


B

regulators		regulatee: LYS2m		
GCN4	LYS14	0	1	2
0	0			
1	0			
2	0	0	3	3
0	1	0	1.14	4.28
1	1	1.69	0	10.55
2	1	9	3	0
0	2			
1	2	24	0	2
2	2	6	0	0

C

regulators		regulatee: LYS2m		
GCN4	LYS14	0	1	2
0	0			
1	0			
2	0	98.6	0	0
0	1	79.9	16.4	0
1	1	0	100	0
2	1	4.35	0.4	95.3
0	2			
1	2	0	96.8	3.2
2	2	0	48.3	47.3



D

Figure 4: **Learning regulation functions.** (A) The optimal transcription regulation function of each lysine biosynthesis pathway enzyme as a function of the states of the regulators GCN4ap and LYS14ap. Each cell presents the state of a regulatee given the states of its regulators GCN4ap (column) and LYS14ap (row). Cell colors indicate the regulatee states. Red (dark gray): state= 2. Dark pink (gray): 1. Light pink (light gray): 0. We show only entries with over 90% confidence. For combinations of regulators states that have lower confidence or were never present in the inferred modes, we leave the corresponding entries of the optimal regulation function undefined. (B). The discrepancy change for each function modification relative to the optimum regulation function of LYS2. Rows: combination of regulators (GCN4ap,LYS14ap) states. Columns: regulatee (LYS2) states. The cell $((x, y), z)$ represents the difference in discrepancy of a regulation function in which $f(x, y) = z$ and all other values are as in the optimal LYS2m function. All values are relative to the discrepancy of the optimal regulation function as shown in A. (C) Confidences for the LYS2 function. Rows and columns are as in (B), values are the percent of times in which the value was learned out of 1000 bootstrap experiments. (D) The accuracy of the independence, Bayesian and MetaReg methods on the lysine biosynthesis pathway. The accuracy is computed by cross validation on all expression conditions and the lysine biosynthesis pathway enzymes.

as regulators. For each variable $v \in L$ and each condition c, we optimized the regulation function of v while fixing the rest of the model and hiding the data of c. We then used the optimized model to infer the mode in condition c without using the observed value of v. Finally, we compared the inferred state of the enzyme variable to the observed one, and counted the total number of correct outcomes (or fractions of outcomes in case the inferred mode was ambiguous and several alternatives existed). Using mRNA expression data only, the accuracy derived in this procedure was 78.3% (Figure 4D).

We compared the performance of Metareg to the following alternative methods: (a) A Bayesian networks [9] with a known structure where GCN4m and LYS4m are the parents of each variable in L. We learned the local probability parameters [11] using non-informative prior. To compute the accuracy, we ran a cross validation test by learning parameters while hiding one condition at a time. The overall accuracy obtained in this procedure was 61.4%, much lower than achieved by *MetaReg*. (b) An independence model: Each regulatee in L has no regulators. We predict the probability of each regulatee outcome as the background distribution of its observations. To compute the accuracy, we ran the same procedure as in (a). The overall accuracy obtained in this procedure was 47.5%. We conclude that the detailed modeling of interactions among proteins, metabolites and mRNAs gives an improved accuracy to our model.

7 Discussion

Models of biological regulation are becoming increasingly complex. The well established biological methodology of model development and expansion (incremental refinement) is facing major challenges with the advent of high throughput technologies and the discovery of more and more regulatory mechanisms. Computational techniques for modeling and learning biological systems are currently limited in their ability to help biologists to extend their models: De-novo reconstruction methods ignore available biological knowledge, and module-based methods do not specify concrete regulation functions. Here we aim at the construction of a computational methodology that combines well with current biological methodologies. MetaReg models can be built for almost any existing biological system, they do not assume complete knowledge of the system, and are flexible enough to integrate diverse regulatory mechanisms. Once built, the model allows easy integration of high throughput data into the analysis of the existing model. The computational tools introduced here can then be used to generate testable and easy to understand biological regulation hypotheses.

Acknowledgment

IGV was supported by a Colton fellowship. AT was supported in part by a scholarship in Complexity Science from the Yeshaia Horvitz Association. RS was supported in part by the Israel Science Foundation (Grant 309/02) and by the EMI-CD project that is funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health", contract number LSHG-CT-2003-503269. "The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability."

References

[1] Z. Bar-Joseph, G.K. Gerber, T.I. Lee, N.J. Rinaldi, J.Y. Yoo, F. Robert, D.B. Gordon, E. Fraenkel, T.S. Jaakkola, R.A. Young, and D.K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21:1337–1342, 2003.

[2] J. L. Crespo and M. N. Hall. Elucidating TOR signaling and rapamycin action: Lessons from *S. cerevisiae*. *Microb. Mol. Biol. Rev.*, 66:579–591, 2002.

[3] P. Dhaseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16:707–726, 2000.

[4] G. Giaever et al. Functional profiling of the *S. cerevisiae* genome. *Nature*, 418:387–391, 2002.

[5] K. Natarajan et al. Transcriptional profiling shows that GCN4p is a master regulator of gene expression during amino acid starvation in yeast. *Mol. Cell. Biol.*, 21:4347–4368, 2001.

[6] S. Even. *Graph Algorithms*. Computer Science Press, Potomac, Maryland, 1979.

[7] A. Feller, F. Ramos, A. Pierard, and E. Dubois. LYS80p of *S. cerevisiae*, previousely proposed as a specific repressor of LYS genes, is a pleiotropic regulatory factor identical to Mks1p. *Yeast*, 13:1337–1346, 1997.

[8] A. Feller, F. Ramos, A. Pierard, and E. Dubois. In *S. cerevisiae*, feedback inhibition of homocitrate synthase isoenzymes by lysine modulates the activation of LYS gene expression by LYS14p. *Eur. J. Biochem.*, 261:163–170, 1999.

[9] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comp. Biol.*, 7:601–620, 2000.

[10] A. P. Gasch et al. Genomic expression programs in the response of yeast to environmental changes. *Mol Biol Cell*, 11:4241–57, 2000.

[11] D. Heckerman, D. Geiger, and D.M. Chickering. Learning baysian networks: the combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft research, 1995.

[12] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, New York, 1972. Plenum Press.

[13] F. Ramos, E. Dubois, and A. Pierard. Control of enzyme synthesis in the lysine biosynthetic pathway of *S. cerevisiae*. *Eur. J. Biochem.*, 171:171–176, 1988.

[14] 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.*, 34(2):166–76, 2003.

[15] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15:281–288, 1995.

[16] A. Tanay and R. Shamir. Computational expansion of genetic networks. *Bioinformatics*, 17:S270–S278, 2001.

[17] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genome-wide data. *Proc. Natl. Acad. Soc.*, 101:2981–2986, 2004.

[18] J. J. Tate, K. H. Cox, R. Rai, and T. G. Cooper. Mks1p is required for negative regulation of retrograde gene expression in *S. cerevisiae* but does not affect nitrogen catabolite repression-sensitive gene expression. *J. Biol. Chem*, 277:20477–20482, 2002.

[19] M.P. Washburn. Protein pathway and complex clustering of correlated mRNA and protein expression analyses in *S. cerevisiae. PNAS*, 100:3107–3112, 2003.

8 Appendix 1: Preprocessing the Data

8.1 Microarray data

Gene expression measurements using cDNA microarrays do not provide absolute mRNA levels in a single condition of interest, but only relative mRNA levels in pairs of conditions. Hence, an experiment gives for each gene the ratios of its mRNA levels in a *test* and *reference* conditions. In order to use such data in our analysis, we first have to transform experiments into absolute mRNA levels in single conditions and then use them to evaluate model discrepancy. This can be easily done if all experiments are using a common reference condition. In this simple case, we may directly use the relative measurements as the observed states in the test condition, due to the fact that all relative measurements are comparable.

In practice, not all experiments use the same reference condition. To handle this case we define the *experiments graph*, a directed multigraph in which vertices represent conditions and arcs represent experiments. For each experiment, we add an arc from the reference to the test condition vertex. Note that condition vertices can be reused. Let $l(v, i, j)$ be the logarithm of the ratio of gene v's levels in condition i and condition j. For gene v, the weight of arc (i, j) is precisely $l(v, i, j)$, as obtained from the microarray in the experiment with test condition j and reference condition i. We wish to compute a normalized log hybridization level (we will refer to it as *level*) for each mRNA variable in each condition. Assume first that the graph is connected. The idea is to fix one vertex in the graph as a common reference and compute the levels of all other conditions relative to it. We then discretize the levels to generate the observed states of each condition.

Our normalization procedure works as follows: First, using prior biological knowledge we fix the levels of the mRNA variables in one condition (the *source condition*). Second, we use a breadth-first search algorithm on the underlying undirected experiment graph in order to handle condition vertices by an ascending distance from the source condition. When reaching a condition i, there must be at least one neighbor whose levels are already determined. We compute the level of each mRNA variable v in the condition i as follows: The *contribution* of an incoming neighbor j is defined as $l(v, j) + l(v, j, i)$ where $l(v, j)$ is the level of variable v in condition j. For an outgoing neighbor j, the contribution is defined as $l(v, j) - l(v, j, i)$. We compute the level of v at condition i by averaging the contributions of all determined neighbors. In case more than one connected component exists, we must fix a common reference in each component and ensure their levels are comparable. Note that if we have absolute measurements on more than one condition in a connected component of the experiment graph, the algorithm can be adapted to take this information into account.

In order to calculate the observed states for datasets (a)-(d) (Section 6.2), we created the experiment graph shown in Figure 5. We used growth of a wild-type strain in standard conditions and YPD medium as our source condition, and fixed its levels to 1 for all genes. We assigned the computed levels in the range $(-\infty,0)$,$[0,2)$ and $[2,\infty)$ to the observed states 0,1 and 2, respectively.

8.2 Protein data

The observed states of proteins in minimal media were obtained by discretizing the concentration levels reported by [19]. We assigned the values in the range $[0,0.5]$,$(0.5,1.5]$, and $(1.5,\infty)$ to the observed states 0,1 and 2 respectively.

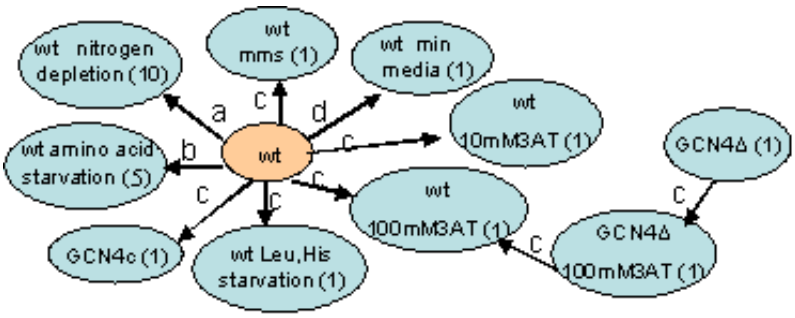


Figure 5: **Experiments graph.** The conditions and experiments used in our analysis. Ovals represent groups of conditions vertices, shown together to simplify graphical representation (the number of conditions represented by each oval is shown in parentheses). Arcs represent differential gene expression experiments and are annotated by their respective publication code: a,b:[10]. c:[5]. d:[19]. wt: wild type in standard conditions and YPD medium.

8.3 Phenotype Data

The observed states of the growth phenotypes were obtained by discretizing the sensitivity scores reported in [4]. Giaever et al. (2002) computed for each strain and environmental condition a sensitivity score, and assigned the sensitivity scores in each generation time a cutoff of significance. Following these cutoffs, we discretized the sensitivity scores in the range [0,10],(10,20], and (20,∞) to observed states 0,1, and 2, respectively for 5 generation phenotypes, and [0,20],(20,100],(100,∞) for 15 generation phenotypes. The growth phenotype was associated with the level of the internal lysine variable in the model.

9 Appendix 2

Proposition 7 *The function optimization problem is NP hard.*

Proof: The decision version of the problem is to determine if there is a function g with $\sum_i D(M(g,v),e^i) \leq L$. We shall reduce 3SAT to it. The idea is to encode a truth assignment of n variables using a function on $\lceil \log n \rceil$ regulators, where each assignment of regulators values encodes one variable and the regulatee value represents the truth value for that variable. For each clause we will add a model variable and a condition, such that the model discrepancy will be zero iff the function encoding the truth assignment will set at least one of the clause literals to its required truth value. For an example of the construction see Figure 6.

We now describe the construction formally. We are given an instance of 3SAT with a set of m clauses $C_1, ..., C_m$, involving the variables $x_1, ..., x_n$. Define first some auxiliary variables: For the j-th literal of clause i, we let a_{ij} equal the index of the variable, and we set $b_{ij} = 0$ if that variable is negated and $b_{ij} = 1$ otherwise. In our example (Figure 6), $C_2 = \neg x_1 \vee \neg x_2 \vee x_3$ and $a_{20} = 1, a_{21} = 2, a_{22} = 3, b_{20} = 0, b_{21} = 0, b_{22} = 1$.

We are now ready to construct the model M. All states are Boolean. The model variables include a) a set of $d_v = \lceil \log_2 n \rceil$ variable pairs r_i, r'_i that constitute a *variable encoding gadget*. b) a variable c_i for each clause C_i and c) a *truth assignment* variable v, to which we will apply function optimization. The variable encoding gadget is built so that each combination of states of its variables corresponds to one of the SAT variables. We introduce a positive feedback loop between r_i and r'_i so that in each mode both can have either 0 or 1 state. The regulation function of each clause variable c_i encodes

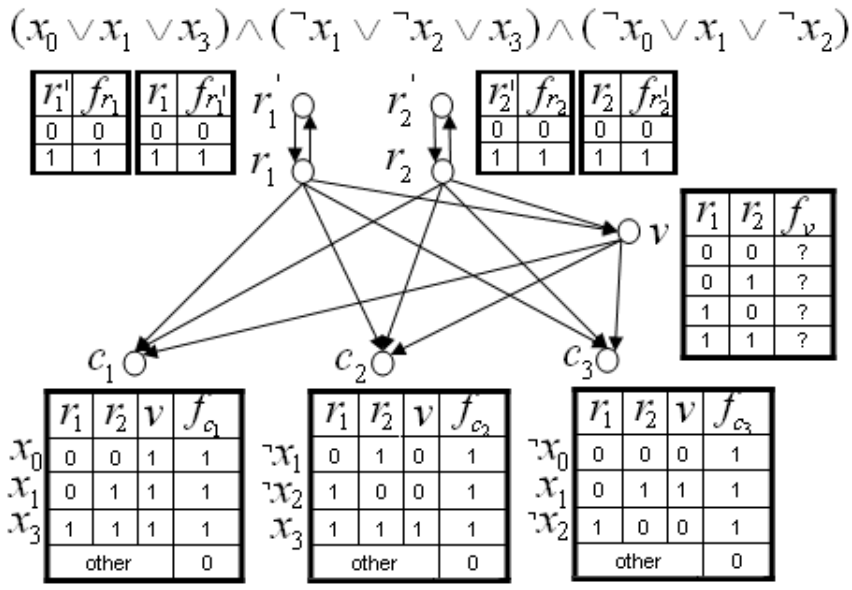


Figure 6: **Construction of a reduction model.** Top: two variable encoding gadgets. Bottom: Three clause variables. Middle: The truth assignment variable whose function is optimized.

the logic of that clause. We let $N(c_i) = (r_1, \ldots, r_{d_v}, v)$ and set f_{c_i} to zero in all but three particular regulators states, corresponding to each of the clause literals, in which the regulatee value is set to 1. The positive regulators state for the j'th clause is determined by the binary representation of a_{ij} in the variable encoding gadget and by requiring v to attain the corresponding b_{ij} value. In other words, f_{c_i} is true only when the variable encoding gadget encodes one of the variables in C_i and v has a value which is compatible with that variable sign in C_i. v's regulators are all the r_i's. To finish the construction, we must specify the set of conditions E. It consists of one condition e^i per clause C_i, in which c_i is observed as 1 and all other variables are hidden. Finally, set $L = 0$.

We claim that there is a function g s.t. $\sum_i D(M(g, v), e^i) = 0$ iff the given instance of 3SAT is satisfiable. We map between truth assignments $T = \{t_j\}$ and functions g^T by setting $g^T(x_1, \ldots, x_{d_v}) \equiv t_j$ where j is the integer with binary encoding $x_1, \ldots, x_{d_v}$. We will prove that all clauses are satisfied by T iff $\sum_i D(M(g^T, v), e^i) = 0$.

Let T be a truth assignment that satisfies the 3SAT instance. Then in each clause C_i there is a variable a_{ij} with sign b_{ij} that is true. Consider the encoding of the index a_{ij} in the variables $r_1, \ldots, r_{d_v}$. By the definition of g^T, the state of v equals $t_{a_{ij}}$, and hence by construction f_{c_i} will equal one, giving zero discrepancy for the condition i. Hence the overall discrepancy is zero.

Suppose conversely that the total discrepancy is zero for the function g^T, and consider the truth assignment T. As the discrepancy for experiment e^i is zero, there is an index a_{ij} whose encoding along with b_{ij} correspond to a value 1 of f_{c_i}. Hence, clause C_i is satisfied. ■

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 13, Number 2, 2006

Pp. 165–181

A Probabilistic Methodology for Integrating Knowledge and Experiments on Biological Networks

IRIT GAT-VIKS, AMOS TANAY, DANIELA RAIJMAN, and RON SHAMIR

ABSTRACT

Biological systems are traditionally studied by focusing on a specific subsystem, building an intuitive model for it, and refining the model using results from carefully designed experiments. Modern experimental techniques provide massive data on the global behavior of biological systems, and systematically using these large datasets for refining existing knowledge is a major challenge. Here we introduce an extended computational framework that combines formalization of existing qualitative models, probabilistic modeling, and integration of high-throughput experimental data. Using our methods, it is possible to interpret genomewide measurements in the context of prior knowledge on the system, to assign statistical meaning to the accuracy of such knowledge, and to learn refined models with improved fit to the experiments. Our model is represented as a probabilistic factor graph, and the framework accommodates partial measurements of diverse biological elements. We study the performance of several probabilistic inference algorithms and show that hidden model variables can be reliably inferred even in the presence of feedback loops and complex logic. We show how to refine prior knowledge on combinatorial regulatory relations using hypothesis testing and derive p-values for learned model features. We test our methodology and algorithms on a simulated model and on two real yeast models. In particular, we use our method to explore uncharacterized relations among regulators in the yeast response to hyper-osmotic shock and in the yeast lysine biosynthesis system. Our integrative approach to the analysis of biological regulation is demonstrated to synergistically combine qualitative and quantitative evidence into concrete biological predictions.

Key words: biological systems, probabilistic modeling, high throughput data.

1. INTRODUCTION

THE INTEGRATION OF BIOLOGICAL KNOWLEDGE, high throughput data, and computer algorithms into a coherent methodology that generates reliable and testable predictions is one of the major challenges in today's biology. The study of biological systems is carried out by characterizing mechanisms of biological regulation at all levels, using a wide variety of experimental techniques. Biologists are continuously refining models for the systems under study, but rarely formalize them mathematically. High-throughput techniques have revolutionized the way by which biological systems are explored by generating massive

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel.

amounts of information on the genomewide behavior of the system. Genomewide datasets are subject to extensive computational analysis, but their integration into existing biological models is currently done almost exclusively manually. To rigorously integrate biological knowledge and high-throughput experiments, one must develop computational methodologies that accommodate information from a broad variety of sources and forms and handle highly complex systems and extensive datasets.

Recent studies on computational models for biological networks have attempted *de novo* reconstruction of a network on genes (e.g., Friedman *et al.* [2000]), used prior knowledge on network topology (e.g., Hartemink *et al.* [2002] and Imoto *et al.* [2004]), or combined transcription factor location and sequence data to learn a clustered model for the genomewide behavior of the system (Bar-Joseph *et al.*, 2003; Segal *et al.*, 2003; Beer and Tavazoie, 2004). Other studies built detailed models manually, utilizing existing biological knowledge (Chen *et al.*, 2000; Covert *et al.*, 2004) but lacked computational methods for model reassessment in light of additional evidence.

In this study, we describe a new mathematical framework for representing biological knowledge and integrating it with experimental data. Our methodology allows biologists to formalize their knowledge on a system as a coherent model and then to use that model as the basis for computational analysis that predicts the system's behavior in various conditions. Most importantly, our framework allows the learning of a refined model with improved fit to the experimental data.

In previous works (Tanay and Shamir, 2001; Gat-Viks *et al.*, 2004), we have introduced the notions of model refinement and expansion and studied it when applied to discrete deterministic models. Here we study these problems in the more general settings of probabilistic models. The probabilistic approach allows us to model uncertainty in prior biological knowledge and to distinguish between regulatory relations that are known at a high level of certainty and those that are more speculative. The probabilistic model also allows us to mix noisy continuous measurements with discrete regulatory logic. Our model expresses diverse biological entities (e.g., mRNAs, proteins, metabolites) and biological relations (e.g., transcription and translation regulation, posttranslational modifications). We formalize our model as a probabilistic factor graph (Kschischang *et al.*, 2001), accommodating undelayed feedback loops which are essential in many biological systems.

Having established our methodology for probabilistic modeling, we develop algorithms for inferring the system's state given partial data. For example, we can infer the activity of proteins given gene expression data. We use inference algorithms as the basis for learning refined regulatory functions. We develop a formulation of the learning problem in our network model, which is based on deterministic hypothesis testing. Our approach to the learning of regulatory models uses regulatory features with clear biological meaning and allows the derivation of p-values for learned model features.

We tested the performance of our algorithms on simulated models and on two complex pathways in *S. cerevisiae*: the regulation of lysine biosynthesis and the response to osmotic stress. In both cases, our models successfully integrate prior knowledge and high throughput data and demonstrate improved performance compared to extant methods. In particular, our results suggest a novel model for regulation of genes coding for components of the HOG signaling pathway and robustly learn logical relations among central transcription factors downstream of the Hog1 kinase. Our results show that integration of prior biological knowledge with high-throughput data is a key step toward making computational network analysis a practical part of the toolbox of the molecular biologist.

The paper is organized as follows: In Section 2 we introduce our mathematical formulation for prior biological knowledge and experimental data. In Section 3, we show how to infer the state of hidden variables. Sections 4 and 5 present our learning methodologies: Section 4 focuses on our discretization scheme and how we propose to learn it. Section 5 presents our mathematical formulation for learning regulation functions and describes a way to assign statistical meaning to the learned functions. Section 6 presents our results on the lysine and HOG pathways. In Section 7, we discuss the advantages and limitations of our approach and outline future research directions.

A preliminary version of this study appeared in the proceedings of RECOMB 2005 (Gat-Viks *et al.*, 2005).

2. MODELING PRIOR KNOWLEDGE AND EXPERIMENTAL OBSERVATIONS

In this section, we present our probabilistic model for a biological regulatory network. We start by defining model variables and formulating prior knowledge on the relations among them. We then incorporate

experimental evidence into the model and show how to combine prior knowledge and experiments into one integrated probability distribution.

2.1. Variables, topology and logic

The biological entities in the system under study are formulated as variables representing, e.g., mRNAs, proteins, metabolites, and various stimulators. We assume that at a given condition, each of the entities attain a logical state, represented by an integer value of limited cardinality. We wish to study regulatory relations (or regulation functions) among variables. Such relations, for example, determine the level of an mRNA variable as a function of the levels of a set of transcription factor protein variables, or the level of a metabolite variable given the levels of other metabolites and of structural enzymes.

In most studied biological systems, substantial prior knowledge on regulatory relations has accumulated. Such knowledge includes direct regulatory interactions, qualitative functional roles (activator/repressor), combinatorial switches, feedback loops, and more. Typically, that information is incomplete and of variable certainty. In order to optimally exploit it, we must model both the relations and their level of certainty. We do this by introducing a distribution on the regulation functions for each variable. This distribution may assign high probability to a single regulation function if our prior knowledge is very strong. At the other extreme, lack of information is modeled by uniform distribution over all possible regulation functions.

We formalize these notions as follows (see Fig. 1A). Let $X = \{X_1, \ldots, X_n\}$ be a collection of biological variables. Let $S = \{0, 1, \ldots, k-1\}$ be the set of logical *states* that each variable may attain. A *model state* s is an assignment of states to all the variables in X. Each variable X_i is regulated by a set of its *regulator* (or *parent*) variables $Pa_i = \{Pa_{i,1}, \ldots, Pa_{i,d_i}\} \subseteq X$. When addressing a particular regulation relation, the regulated variable is also called the *regulatee*. Lower case letters will indicate state assignments of the corresponding upper case variables. For example, given a model state s, x_i^s is the state of X_i, pa_i^s is the assignment of the set Pa_i. The *regulatory dependency graph* is a digraph $G_R = (X, A)$ representing direct dependencies, i.e., $(X_u, X_v) \in A$ iff $X_u \in Pa_v$ (G_R is sometimes called the *wiring diagram* on the *topology* of the model). The graph can contain cycles. The *regulation function prior* for a variable X_i is formulated as our belief that the variable attains a certain state given an assignment to its parents Pa_i. It is represented by the conditional probabilities θ^i:

$$\theta^i(X_i, Pa_i) = Pr(X_i | Pa_i) \tag{1}$$

2.2. From measurements to logical states

In practice, biological experiments provide noisy observations on a subset of the variables in the system. The observations are continuous, and we do not know in advance how to translate them into logical states. We thus introduce a set of real-valued *sensor variables* $Y_1, \ldots, Y_n$ and *discretizer distributions* $\psi^i(X_i, Y_i)$ that specify the joint distribution of a discrete logical state of X_i and the continuous observation on Y_i. In this work, we shall use mixtures of Gaussian (Fig. 1B) to model ψ^i, but other formulations are also possible. Note that we chose to formulate the relations between the logical and sensor variables as joint rather than as conditional probabilities $P(Y_i | X_i)$.

In addition to providing partial observations on model variables, experiments are performed in a specific environment and may possibly perturb some of the regulation functions in the system (for example, by knocking out or overexpressing some genes). We model these by fixing the values of logical variables that correspond to the environment and by changing the regulation function priors (the θ factors) to reflect the perturbations.

2.3. The factor graph network model

Our model is defined by a joint distribution over a set of logical (X) and sensor (Y) variables. The distribution is constructed as the product of the *factors* θ^i, ψ^i, such that

$$Pr_M(X, Y) = \frac{1}{Z} \prod_i \theta^i(X_i, Pa_i) \psi^i(X_i, Y_i) \tag{2}$$

where Z is a normalization constant.

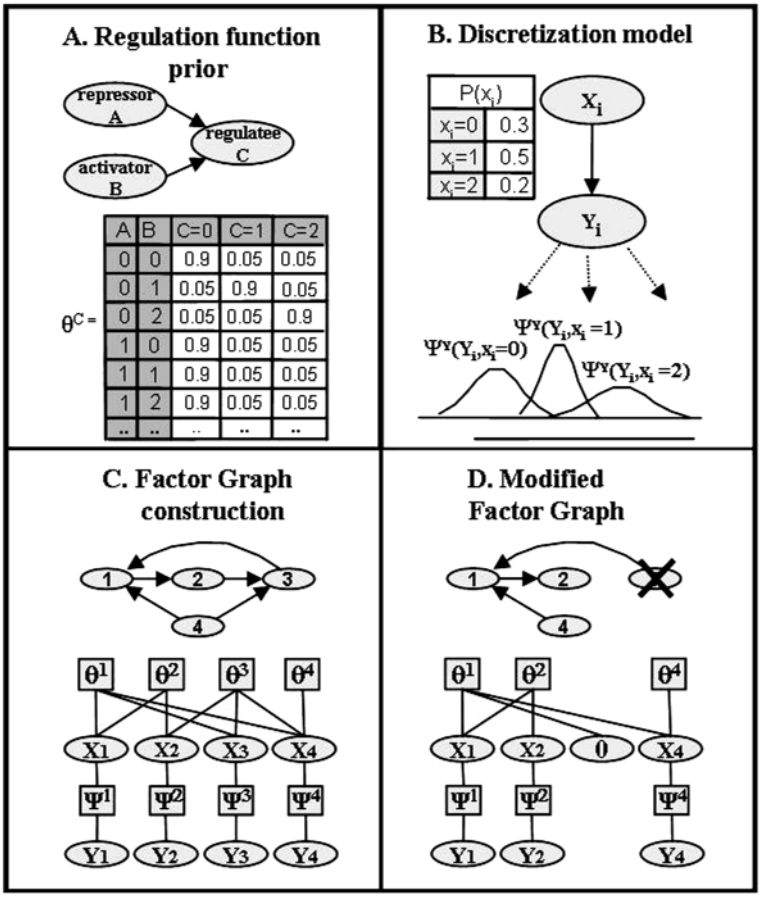


FIG. 1. An overview of the factor graph network model. (**A**) Knowledge on the logical regulation functions is formalized as conditional probabilities. (**B**) Continuous measurements and logical states are linked by joint discretizer distributions. (**C**) A possibly cyclic network structure G_R (top) is transformed into a factor graph (bottom), using the regulation function priors and the discretizers' distributions. (**D**) Given an experiment in which gene X_3 is knocked out, the model is modified accordingly by fixing the state of X_3 to zero and eliminating the corresponding factors Θ^3, Ψ^3.

We can represent our joint distribution using a probabilistic factor graph (Kschischang *et al.*, 2001) which explicitly expresses the structure of the joint distribution?s factorization. Factor graphs are widely used probabilistic graphical models that were originally applied to coding/decoding problems (for a different application of factor graphs in computational biology, see Yeang *et al.* [2004]). A factor graph is a bipartite graph associating variable nodes (in one side of the graph) with factor nodes (in the other side of the graph). We add an edge between a variable x and a factor f_j if the scope of f_j contains x. In our case (Fig. 1C), the factor graph has a variable node for each logical and sensor variable (X, Y) and a factor node for each function θ^i, ψ^i. A modified factor graph representation, matching a perturbation experiment, is shown in Fig. 1D. We call this formulation a *factor graph network (FGN) model*. Note that our formulation is undirected although part of our model (the sensor variables) represent conditional probabilities. Although it is in principle possible to use hybrid models (e.g., chain graphs Buntine [1995]) and maintain the directionality information in the model, for our purpose here, the undirected formulation suffices.

When the dependency graph G_R is acyclic, our FGN model is equivalent to a Bayesian network on the variables X_i and Y_i, constructed using the edges of G_R and additional edges from each X_i to the corresponding Y_i. This can be easily seen from (2) by noting that in the acyclic case $Z = 1$ (the proof is as in Bayesian networks theory, e.g., Pearl [1988]). When the model contains loops, the situation gets more

complicated. For example, we note that according to the FGN model, $Pr_M(X_i|Pa_i)$ does not necessarily equal the original beliefs $\theta(X_i, Pa_i)$.

We note that the semantics of the prior θ^i distributions is different than that used in previous works (e.g., Friedman *et al.* [2000]), where they served as conditional probabilities on the values of the variables in a probabilistic setting. Instead, we assume that the true model *deterministically* determines X_i given its parents, but we are not sure which deterministic rule applies, and therefore what value X_i will attain. Regulation functions approximate an underlying biochemical reaction whose exact parameters are usually not known. The regulatory process is stochastic at the single cell level, but the parameters of the reaction equations governing it are deterministic. When we observe a large ensemble of cells in a high-throughput experiment, we average millions of stochastic processes and in theory should obtain an almost deterministic outcome or a superposition of several deterministic modes. Such deterministic outcome is obscured by significant experimental noise, so a practical modeling approach may assume uncertainties on deterministic logic and noisy observations. In the future, given measurements at the single cell level, the θ distributions may be applicable to describe the inherent stochasticity of some biological switches.

3. INFERENCE

In this section, we discuss the inference problem in the FGN model. Each experiment provides partial information on the value of model variables. Typically, a subset of the sensor real-valued variables are observed in each experiment (for example, mRNA variables are determined in a gene expression experiment), and the model is modified according to some perturbations at the appropriate condition (compare Fig. 1D). The inference problem seeks the computation of the distribution of hidden (unmeasured) variables given the experimental data and the model.

There are two types of inference problems we shall deal with. The first problem (*marginal inference*) is to compute *posterior distribution*s for a single hidden variable. For example, given a gene expression profile D (specifying observations on all mRNA sensor variables), we may wish to compute the marginal $P(X_i|D)$ of a protein variable or a certain metabolite. The second problem is to compute the likelihood $P(D)$ of the observed data D. Inference in graphical models is an NP-hard problem (Cooper, 1990) that was extensively studied. We explored the effects of our model's specific characteristics on the performance of three inference algorithms. Specifically, we implemented a Gibbs sampler, the loopy belief propagation algorithm, and a structure-based instantiation inference algorithm.

The *Gibbs sampler* is a naive MCMC algorithm (MacKay, 1998) that performs a random walk over the space of model states, based on sampling from local distributions. To perform Gibbs sampling, we convert the FGN model to the equivalent Bayesian network as described by Yedidia *et al.* (2004). In our model, sampling is done only for the logical variables (unobserved sensors do not affect marginal posteriors of the logical variables, since they are integrated to 1).

The loopy belief propagation (LBP) algorithm belongs to a popular class of algorithms (Yedidia *et al.*, 2004) which approximate the posterior distribution assuming certain decomposition over independent variables or clusters of variables. Algorithms from this class include LBP, mean field, and their generalizations. The LBP algorithm for the FGN model (implemented as described by Yedidia *et al.* [2004]) is a message-passing procedure that is guaranteed to reach an exact solution for acyclic models and was reported to perform well in some cases of cyclic models.

We also developed an instantiation-based inference algorithm that exploits the known dependency structure of the model and builds on ideas from the deterministic network model (Gat-Viks *et al.*, 2004). Briefly, recall that a deterministic (possibly loopy) network model is defined by a set of deterministic regulation functions (one for each variable) and that such a network may attain a limited number of steady states (or *modes*) in which the value of each variable is correctly determined by its regulation function and the values of its regulators. Also recall that in an acyclic model (or in a loopy model in which the values of the variables in a feedback set are fixed), the mode is uniquely determined (if one exists). We can therefore search for modes in a deterministic network by analyzing the underlying topology, identifying a feedback set, and enumerating over all value assignments for it. The modes instantiation (MI) algorithm first builds a deterministic network model by taking, for each variable, the maximum likelihood regulation function (using the prior θ^i and breaking ties arbitrarily). It then identifies a feedback set in G_R and computes

the appropriate set of modes. The algorithm next computes the likelihoods of each mode in the original probabilistic model, possibly optimizing it using a greedy algorithm. The results of this procedure are a set of model states with locally optimal likelihoods. For models that are close to being deterministic in many of the variables, such set of modes may represent a significant chunk of the total likelihood of the model given the data. The algorithm therefore approximates the posterior distribution as a mixture of modes, weighted by their likelihoods. Since the number of modes may be small in practice, the algorithm adds to the set of locally optimal states an additional small set of states derived using the Gibbs sampler, generating a more smooth approximation for the posterior. The MI algorithm constructs a tractable estimation of the joint posterior which can be used both for computation of marginal posteriors and for estimation of the full probability $P(D)$. There are no guarantees for the quality of this approximation, but our empirical studies suggest that the algorithm works well in practice, probably due to the nature of models we use (strong priors on many of the regulation functions).

We tested the three inference algorithms on a simulated model (see Fig. 2). We constructed simulated FGN models by starting from a deterministic model and randomizing it. We use a *prior strength* parameter α to construct θ functions that assign probability α for the anticipated deterministic function outcome and $\frac{1-\alpha}{k-1}$ to other values. For a detailed description of the simulation, see our website *www.cs.tau.ac.il/~rshamir/fgn/*.

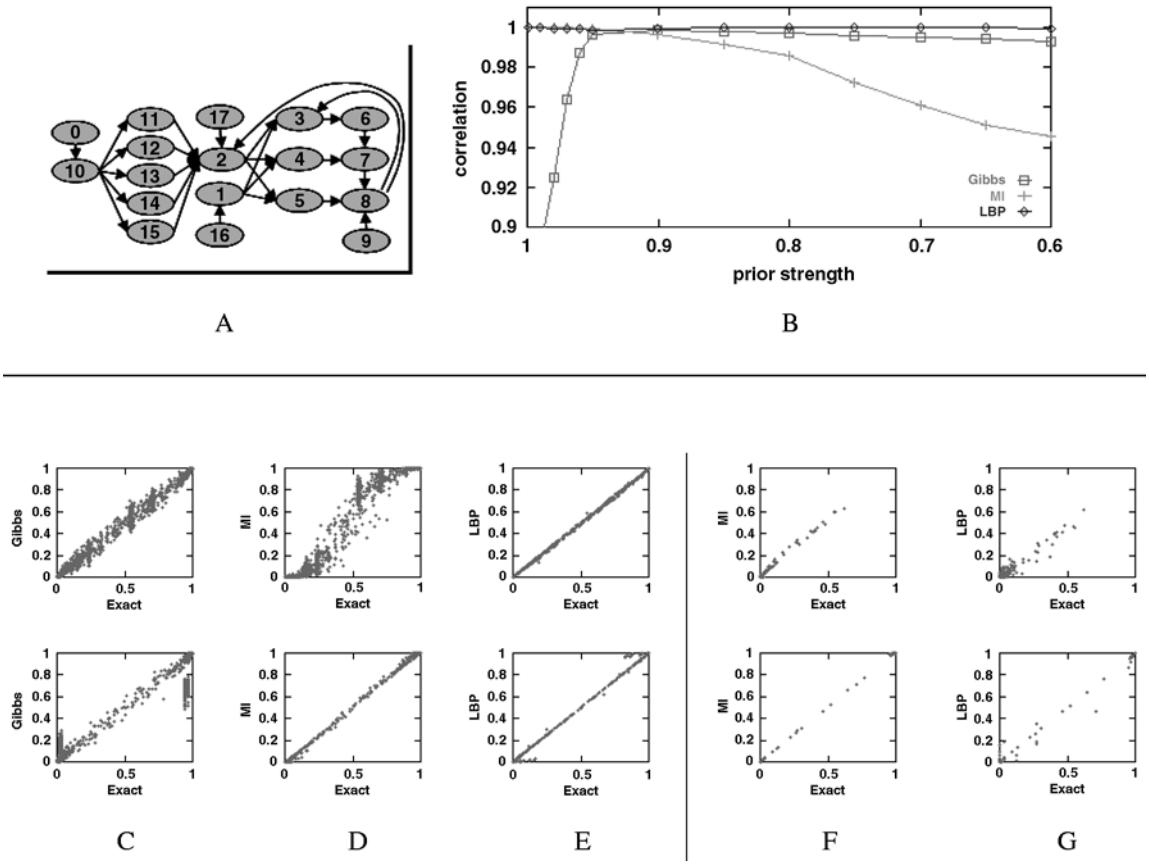


FIG. 2. Performance of different inference algorithms on a simulated model. Performance is measured by the correlation of the inferred and the exact posterior distribution. (**A**) The dependency graph G_R of the simulated model. (**B**) Effect of prior strength on inference accuracy. Y axis: the correlation of the inferred and exact marginal posteriors. X axis: prior strength (α). For strong priors, LBP and MI give a good approximation for the posterior, while the accuracy of the Gibbs sampler is low. As priors get weaker, the performance of MI deteriorates, indicating that the mixture of deterministic states is a poor approximation for the posterior in these cases. (**C**, **D**, **E**) Detailed correlation of inferred and exact marginal posteriors for $\alpha = 0.7$ (top) and $\alpha = 0.97$ (bottom). (**F**, **G**) Detailed correlation of inferred and exact joint posteriors for $\alpha = 0.7$ (top) and $\alpha = 0.97$ (bottom). We see that MI outperforms LBP when comparing the joint posteriors.

We explored the behavior of the different algorithms as a function of the prior strength α using the correct posterior as the reference. Models with α near 1 represent very good knowledge on the system under study. Models with α near $\frac{1}{k}$ represent complete lack of knowledge. We first tested the accuracy of inferring marginal posteriors. Figures 2B,C,D,E indicate that for estimating marginal posteriors, LBP outperforms the other two algorithms (and also the mean field algorithm and a simple clustered variational algorithm [Jaakkola, 2001], data not shown). When the prior is strong, MI provides comparable accuracy. We also wished to test the quality of inferring joint posterior distributions. Joint posteriors cannot be computed directly from LBP, and thus are estimated by multiplying marginal posteriors (assuming independence among the variables). For the MI algorithm, we applied the approximation of the posterior distribution using a mixture of locally optimal states. The results (Figs. 2F,G) confirm that for models with loops and strong prior knowledge, the approximation using the MI algorithm performs better, exemplifying the limitations of the posterior independence assumptions. Overall, we prefer using LBP to infer marginal posteriors and MI to approximate the joint posterior distribution.

4. LEARNING DISCRETIZERS

Adequate transformation of continuous measurements into logical states (i.e., discretization) is essential for the combined analysis of experimental data and a model representing extant biological knowledge. There are several alternative approaches to discretization. In most previous works on discrete models (e.g., Friedman *et al.* [2000] and Gat-Viks *et al.* [2004]), discretization was done as a preprocess, using some heuristic rule to map real-valued measurements into discrete states. In this approach, the rule must be determined and tuned rather arbitrarily, and typically all variables are discretized using the same rule. Here we propose a different approach to discretization. As in the FGN model the discretization is an integral part of the model, the dependencies between the discretization schemes and regulation function priors are fully accounted for. It is thus possible to (a) define different discretization scheme for different variables and (b) apply standard learning algorithms to optimize the discretization functions used. Given a logical function prior and experimental evidence D, we learn the discretization functions ψ^i using an EM algorithm. We initialize all ψ^i using any heuristic discretization scheme. In each EM iteration, we infer the posterior distributions for each of the variables X_i in each of the conditions and then reestimate the ψ^i mixtures using these posteriors, by computing the Gaussians sufficient statistics $E(Y_i|X_i = j, D)$, $V(Y_i|X_i = j, D)$. The new ψ^i distributions are used in the next iteration, and the algorithm continues until convergence.

The FGN model thus provides a very flexible discretization scheme. In practice, this flexibility may lead to overfitting and may decrease learnability. One can control such undesired effects by using the same or few discretization schemes on all variables. As we shall see below, on real biological data, variable specific discretization outperforms global discretization using a single scheme and is clearly more accurate than the standard preprocessing approach.

5. LEARNING REGULATION FUNCTIONS

Given an FGN model and experimental evidence, we wish to determine the optimal regulation function for each variable and provide statistical quantification of its robustness. We assume the parameters of the logical factors in the FGN model represent our prior beliefs on the logical relations between variables and attempt to learn by confirming beliefs, deciding whether a certain regulator assignment gives rise to a certain deterministic regulatee assignment.

5.1. Formulating the learning problem

We focus on the regulation of some variable X_i and attempt to learn a single deterministic feature in the model: the value of X_i given a fixed parents value assignment pa_i^s. Define h_j as the FGN model derived from M by setting $\theta(j, pa_i^s) = 1$ and $\theta(j', pa_i^s) = 0$ for $j' \neq j$ and keeping all other model parameters at their original values. We define the learning problem in our model as selecting the maximum likelihood h_j.

To that end, we shall have to compute the likelihood of the data given each of the h_js, a difficult problem when we have hidden variables even on an acyclic model.

The likelihood of the data, given a model, is approximated by the inference algorithms described above. Recall that we can approximate the full probability using a small number of high-probability modes (using, e.g., the MI algorithm). While this may be a crude approximation, our empirical analysis shows that it is still adequate (see below). Importantly, the likelihood of each h_j takes into account all our prior knowledge on regulation functions and experimental observations.

We note that our approach can be viewed as standard Bayesian learning, using a prior that assumes that the only possible regulation functions are the deterministic ones. We have chosen to represent the learning process as selection of the maximum likelihood discrete hypothesis for two reasons: First, this sharp prior helps us define the semantic of the features that we learn (e.g., activation/repression). Second, it allows standard statistical tools (e.g., likelihood ratio testing) to be applied, so that p-values of learned regulation rules can be derived. In the future, when single cell measurements are available, and when models that explicitly express the stochasticity of regulatory switches are developed, other types of priors may be more appropriate.

5.2. Statistical evaluation

To assign statistical meaning to the learning procedure, we use two methods: bootstrap and likelihood ratio testing. In the bootstrap method, we repeatedly select random subsets of conditions from the original data D and for each one perform the learning procedure. We count the number of times each h_j was selected as the maximum likelihood model and define the feature *robustness* as the fraction of times it was selected. Bootstrap is in widespread use in cases where sampling from the background distribution is impossible or very difficult. In our case, approximated sampling from $Pr(D|h_j)$ is possible given our representation of the posterior landscape as a mixture of modes. We can thus try to directly perform a likelihood ratio test and derive p-values for the learned features.

In a likelihood ratio test, we test the null hypothesis H_0 against the alternative hypothesis H_1. The test statistic is the ratio $\lambda = \frac{max_{h_i \in H_0 \cup H_1} Pr(D|h_i)}{max_{h_i \in H_0} Pr(D|h_i)}$. We decide to reject the null hypothesis (and accept H_1) if an observed ratio λ' is too high and assign this decision a significance level by computing a p-value $pr(\lambda \geq \lambda'|H_0)$. Therefore, the distribution of λ given H_0 must be estimated.

In our case, we fix j and define $H_1 : h_j$, $H_0 : \cup_{k \neq j} h_k$. To estimate the distribution $p(\lambda|H_0)$, we generate samples from the distribution $Pr(D|H_0)$, compute the corresponding λs, and reconstruct the λ distribution. The main problem is therefore the sampling of datasets D. When sampling, we take into account the

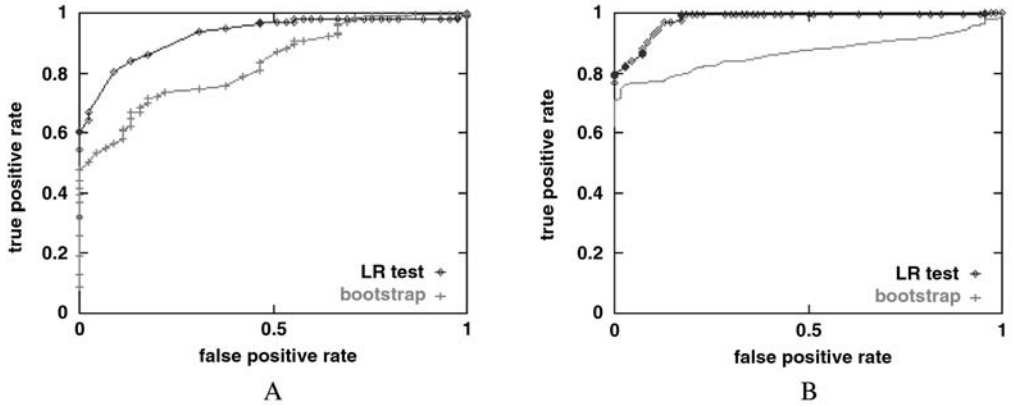


FIG. 3. Accuracy of learning regulation functions. Each figure is a ROC curve (X axis: false positives rate, Y axis: true positives rate) for learning the functions in a simulated model using bootstrap and likelihood ratio test for determining the significance of learned features. LR-test (bootstrap) curves were obtained by varying the p-value (robustness) between 0 and 1 and for each value, averaging over the true positive rates for all variables in the model. Results are shown for learning from 15 (**A**) and 80 (**B**) conditions and represent the average across all model variables. The accuracy of the likelihood ratio test method is consistently higher.

model H_0 that was modified according to the perturbations in each of the experiments (Fig. 1D). We do this as follows: for each of the conditions in the original dataset, we form the modified model according to the experiment. We then apply the MI algorithm to the modified model, without any observation on Y variables, and compute the set of posterior modes for the X variables. These modes represent logical model states that are probable given the experimental conditions. We then generate a sample by (a) selecting a mode from the set of posterior modes, weighted by their likelihood, and (b) generating observations on Y variables using the model discretizer distributions ψ. Our procedure therefore generates a random sample of conditions for a true H_0 model in which the corresponding experimental perturbations were performed.

We analyzed the performance of the bootstrap and likelihood ratio test methods by learning features in our simulated model (see Figure 2A and our website *www.cs.tau.ac.il/∼rshamir/fgn/* for details). Figure 3 shows ROC curves for learning in the simulated model using 15 and 80 conditions. We see consistently better accuracy when using the likelihood ratio tests, probably due to better resolution of features that are nearly ambiguous given the data. While bootstrap has the advantage of not assuming an approximation to the full probability of the data, the likelihood ratio test is more accurate when the posterior can be reasonably approximated.

6. RESULTS ON BIOLOGICAL DATA

In order to test the applicability of our methods to real biological systems, we constructed models of two important yeast pathways, the Hog1 MAPK pathway, which mediates the yeast response to hyperosmotic stress, and the lysine intake and biosynthesis pathway. For each of the models, we performed an extensive literature survey in order to construct the initial model of the system (for the lysine system, our previously developed deterministic model [Gat-Viks *et al.* [2004] was the main source). We collected published experimental data on each of the models.

The HOG model is an acyclic model with 50 variables (outlined in Fig. 4). The lysine biosynthesis model contains 140 variables, 28 of which are involved directly in feedback loops or in the biosynthesis regulation. Figure 5 illustrates only this part of the model. The full topology of the model appears in

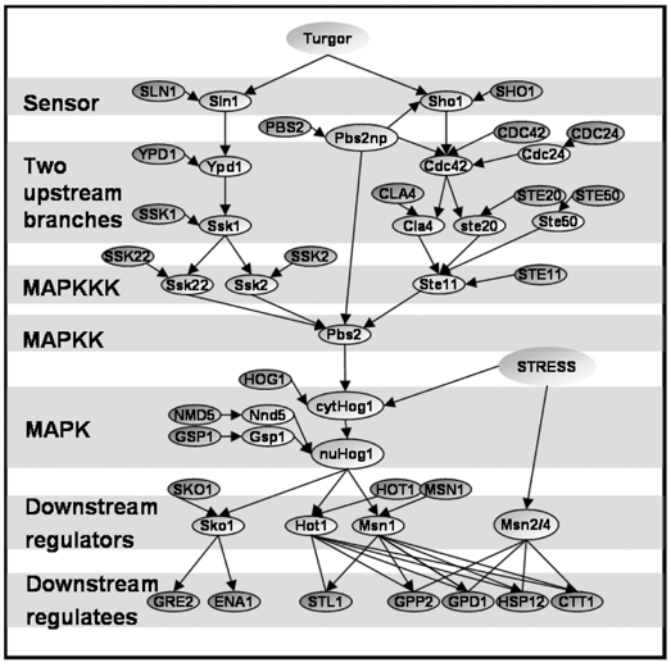


FIG. 4. Topology of the HOG model. The mRNA variable names are capitalized; protein variable names appear with initial capital letters. Turgor and stress are stimulator variables; cytHog1: cytoplasmic Hog1; nuHog1: nuclear hog1.

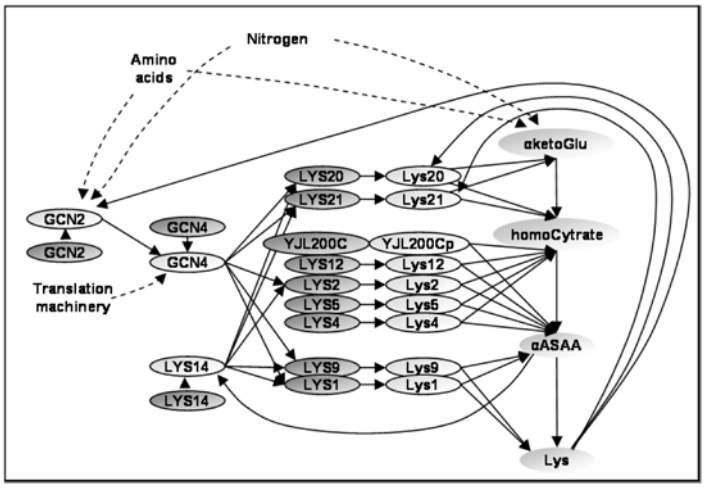


FIG. 5. Partial topology of the lysine biosynthesis model. The mRNA variable names are capitalized; protein variable names appear with initial capital letters. Metabolites are shown as unoutlined ovals. Amino acids and nitrogen transport modeling (110 variables) and translation machinery (10 variables) are not shown.

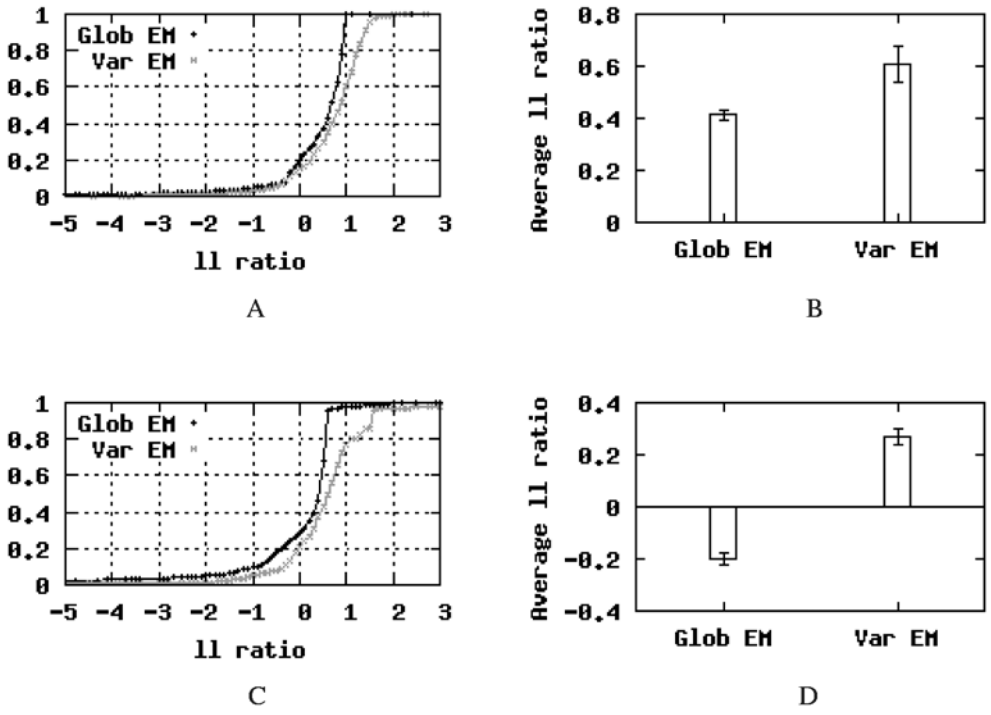


FIG. 6. Learning discretization distributions. Cross validation results for alternative methods for estimating the discretization functions ψ^i in the HOG (**A**, **B**) and lysine model (**C**, **D**). Glob EM: optimized single common discretization function. Var EM: optimized variable specific discretization. (**A**, **C**) Cumulative distribution of log likelihood (ll) ratios comparing each of the two discretization methods to the global predefined discretization scheme. (**B**, **D**) Average ll ratios for the two methods. Bars indicate the predicted standard deviation of the averages.

Gat-Viks *et al.* (2004). The complete description of the models, including the regulation functions, can be found at our website.

We collected published experimental data on each of the models. The data consisted of 129 conditions for the HOG model (O'Rourke and Herskowitz, 2004), and 23 conditions (cf. Gat-Viks *et al.* [2004]) for the lysine model. Differential measurements from cDNA microarrays were transformed into absolute values as described by Gat-Viks *et al.* (2004). In both models, we used three-valued logical variables, with values 0,1,2 corresponding to low, intermediate, and high levels. We used prior strength $\alpha = 0.9$ for all regulation functions in both models.

6.1. Learning discretization

The FGN model couples continuous measurements and discrete states via the discretizer distributions ψ^i. We tested our ability to learn the functions ψ^i by performing cross validation using gene expression data for the HOG and lysine models.

We used cross validation to compare three alternatives: (A) a single common predefined mixture of Gaussians, (B) using the EM algorithm described in Section 4 to learn a single common maximum likelihood ψ distribution, and (C) applying an unconstrained EM to learn variable specific ψ^i-s.

Cross validation was done as follows. For each condition, we used one of the above methods to learn the ψ distributions, using all data excluding that condition. We then iterated over all the model's variables. For each variable v, we hid its observation in the omitted condition and inferred its posterior distribution using the trained ψ's. Finally, we computed the likelihood of v's observation given the posterior.

Figure 6 shows the results of the cross validation on the HOG and lysine models. We present the distribution and the average log likelihood ratio of each of the methods B and C to the predefined discretization (method A). This comparison allows us to view the results in terms of the generalization capabilities of the optimized discretizers: negative log likelihood ratios represent cases where the refined discretization resulted in overfitting. Positive log likelihood ratios represent successful generalizations. We conclude that in both models, incorporating the variable specific discretization into the model improves performance for about 80% of the cases and also improves the average log likelihood ratio. In both cases, variable specific discretization outperforms the optimized single common discretization scheme. Interestingly, in the case of the lysine model, the common discretization scheme performs worse than the predefined discretization, as indicated by its negative average log likelihood ratio (Figure 6D).

6.2. Biological analysis of the HOG model

The response of yeast to hyperosmotic stress is mediated through two parallel MAPK upstream signaling branches, the multitarget MAP kinase Hog1 and an array of transcription factors that coordinate a complex process of adaptation by transient growth repression and modifications to glycerol metabolism, membrane structure, and more (Hohmann, 2002). Two key regulators in this response are regulated also by the general stress response pathway. We have constructed an FGN model that represents known regulatory relations in the HOG system (Fig. 4) and used it to study the transcriptional program following treatment by variable levels of KCl (O'Rourke and Herskowitz, 2004). The data we used contained observations of all the mRNA variables in the model and assignments of fixed values for the logical variables describing experimental conditions (general stress and turgor pressure). To test the prediction accuracy of the prior model, we applied the LBP inference algorithm to estimate the marginal posteriors of all logical variables. We summarize the model predictions in the *discrepancy matrix* shown in Fig. 7. The discrepancy matrix shows the correspondence between model predictions and experimental observations for each single variable under each condition. Essentially, the discrepancy matrix is the result of a leave-one-out cross validation procedure. To generate it, we examine each sensor variable Y_i in each condition. We infer the marginal posterior distribution of Y_i given the observations on all other variables (except Y_i) and compute the expected value and the probability of Y_i observation. We present the difference between the expected values and the observations in a color-coded matrix.

The discrepancy matrix reveals several important discrepancies between the current model for osmoregulation and the microarray experiments we analyzed. We discuss here briefly two major trends. The first trend affects a group of genes coding for proteins participating in the MAPK signaling cascade (*SSK1, SHO1, STE20, PBS2, CDC42, HOG1,* and more). These genes are repressed during the peak of

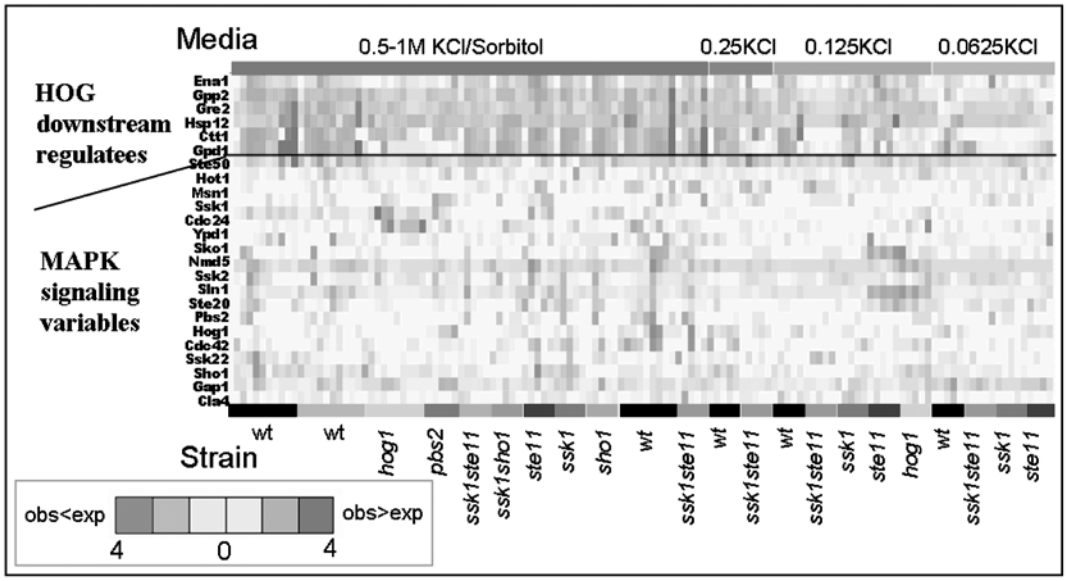


FIG. 7. Discrepancy matrix for the HOG model on data from O'Rourke *et al.* Rows correspond to mRNA variables and columns to experimental conditions from O'Rourke and Herskowitz (2004). Top bars indicate conditions in variable levels of osmotic shock. Bottom colored bars indicate groups of experiments with the same wild type or knockouts at different time points. In treatments of 0.0625 M, 0.125 M and 0.25 M KCL, each group is spanning five time points over 15, 30, and 40 minutes, respectively. In 0.5 M and 1 M, short (long) bars indicate 5 (10) experiments over 40–180 minutes. Dark cells indicate observations that are lower or higher than the expected prediction. Color intensity is proportional to minus the log likelihood of the observation.

the osmo-regulation program (10–30 minutes after treatment with 0.5 M KCl, around 60 minutes in the 1 M KCl treatment). This repression is not reported in the current literature. We hypothesize that as part of the adaptation to high levels of osmotic pressure, yeasts may reduce the sensitivity of the Hog1 signaling cascade, by slowing down the production of some central components in it.

A second group of discrepancies involves genes that are targets of the Hog1 downstream transcription factors. These genes include Sko1, Hot1, and Msn1 (Proft and Serrano, 1999; Rep *et al.*, 1999; Rep *et al.*, 2000) (Fig. 4). In many cases, the literature does not specify the logical relations among the regulators and each of their regulatees, and this lack of knowledge is manifested as discrepancies.

We used our model learning machinery to refine the regulatory logic for several model variables that are known to be affected by Hog1-downstream regulators. Figure 8 shows examples of logical relations

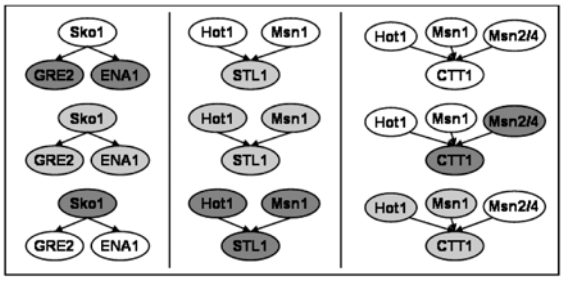


FIG. 8. Learning in the HOG model. Examples of model features learned by the FGN methodology. We show logical relations that were learned with significant p-values. Each graph depicts the regulation of one regulatee given the particular states of its regulators. Variable states are indicated by node colors: white—0, light gray—1, dark gray—2.

learned. First, we were able to learn the known repressive role of Sko1 in the regulation of *GRE2* and *ENA1* (Proft and Serrano, 1999). We learned three model features that associated high levels of the mRNA variables of these two genes with low state of the inferred Sko1 regulator state, and vice versa. The expression of the *SKO1* gene during osmotic stress is static, and the correct regulation function could only be learned given the inferred Sko1 protein activities. These inferred activities take into account, in addition to the mRNA measurements, the entire model and its regulatory functions. We also learned the regulation of *STL1*. That regulation is reported to be completely dependent on Hot1 and Msn1 (Rep *et al.*, 2000), but the literature does not clarify the logical relations among them. Our results show that although these two regulators have a positive effect on *STL1* expression, the gene can be induced even when both regulators lack any activity. We can thus hypothesize that a third factor is involved in *STL1* regulation. A third, more complex regulation function associates the Hog1 specific regulators Hot1, Msn1 and the general stress factor Msn2/4 into a single program controlling several genes. Our model contains only four representatives of a larger regulon: *GPP2*, *GPD1*, *HSP12*, and *CTT1* (Rep *et al.*, 1999). Similar results as for CTT1 (Fig. 8) were obtained also for the other three regulatees (data not shown). Our results indicate that the two signaling pathways (the HOG cascade and the general stress pathway) act in parallel, and each of the pathways can induce the regulon in the absence of activity from the other.

6.3. Biological analysis of the lysine biosynthesis model

Figure 5 shows the core of the lysine biosynthesis system in the yeast *S. cerevisiae*. It includes a linear metabolic pathway from α-ketoglutarate through αAASA to lysine, the catalyzing enzymes of the metabolic reactions (Lys1,2,9,12,20,21, and YJL200C) and their transcription control via the transcription factors Gcn4 and Lys14. Gcn4 activity is regulated during transcription, and Lys14 is influenced by the αASSA positive feedback loop, sensing the lysine biosynthesis flux. Additional feedback loops are the general nitrogen control regulation mediated by Gcn2 and the lysin negative regulation on Lys20 and Lys21. The full model includes also amino acids and ammonium (NH3), which represent the environmental conditions enforced on the system, and their transport into the cell by specific permeases (see Fig. 5 and *www.cs.tau.ac.il/∼rshamir/fgn/* for a full topology and logic).

We now wish to demonstrate the power of the feedback modeling and test the advantage of our method over former methods. The performance of the method is measured here by the capability to learn real regulation functions from real data. We thus apply cross validation in the lysine model and compare the performance of our approach to the deterministic model approach and to a naive Bayesian approach. The deterministic model approach (Gat-Viks *et al.*, 2004) learns a deterministic regulation function by optimizing a least squares score. It assumes a prior model that is 100% certain and solves the deterministic analog of the inference problem to enable the learning of a regulation function from partial observations. To allow comparison of the deterministic model with the current one, we transformed its discrete predictions into continuous distributions using predefined Gaussians. The same discretizers were used in the other two models, in order to ensure that differences in model performance were not due to the discretization. In the naive Bayesian approach, we assume that the topology of a Bayesian network over the observed variables (the mRNAs in our case) is given, and we learn the conditional probabilities of each variable separately given its regulators using complete data. The learning problem in this case is trivially solved by building a frequency table. Learning in the FGN model was done given the probabilistic function priors θ^i. We used the hypothesis testing procedure described above to repeatedly attempt the learning of regulation function features. For a variable with m regulators, we have k^m such features corresponding to each assignment of states to the regulators. For each feature, and given a p-value threshold (we used 0.01), our learning algorithm may or may not be able to decide on the correct regulatee outcome. We update the regulation function to reflect a strong θ for the feature ($\alpha = 0.99$) where a decision was made and a uniform distribution for θ where no decision could be made. We iterate the learning process until no further improvement is possible and report a regulation function in which only a fraction of the features are determined.

To perform the cross validation we repeatedly selected a variable and set its prior θ^i to the uniform distribution. We removed one condition from the dataset, learned the variable's regulation function, and used it to compute the posterior of the variable, given the omitted condition without the observation for the test variable. Figure 9 depicts the log likelihood ratio distribution for the three methods (compared to a uniform

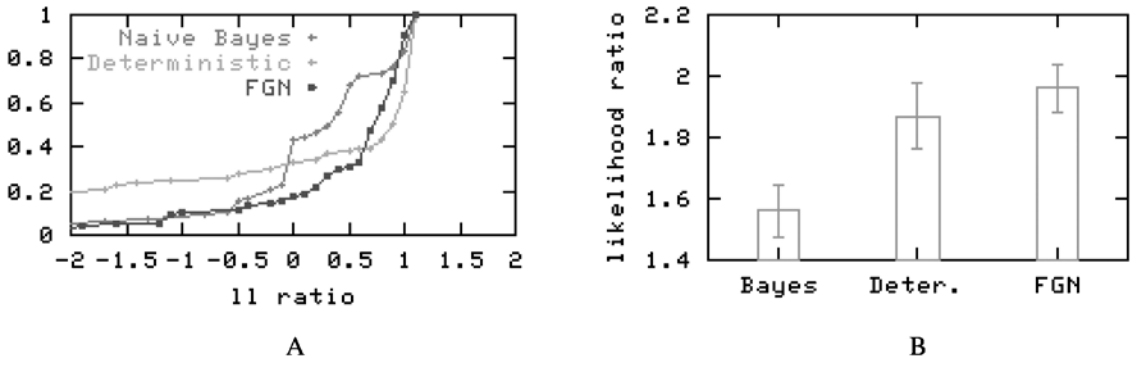


FIG. 9. Performance of different methods for learning regulation functions on the lysine model. Cumulative distributions (**A**) and averages (**B**) of the log likelihood (ll) ratio for cross validation in the lysine model using three methods for learning regulation functions: A naive Bayesian method, assuming the network topology, a deterministic learning scheme as in Gat-Viks *et al.* (2004), and learning using the FGN model. Bars indicate the predicted standard deviation of the averages.

prior model). We see that the FGN model improves over the other two methods. Detailed examination of the distribution reveals that the probabilistic model makes half as many erroneous predictions (negative log likelihood ratios) as its deterministic counterpart, probably due to its ability to evaluate statistically the learning predictions and thus avoid false positive predictions. Both the deterministic and probabilistic methods make good use of the additional knowledge, formalized into the model logic, to obtain better results than the naive Bayesian approach.

Figure 10 shows an example of how the formalized biological knowledge might improve the learning performance. In order to learn the regulation of the biosynthesys enzymes (e.g., LYS1,9,20) by their regulators Gcn4 and Lys14, our model infers the protein levels of the regulators and uses it as the basis for the learning process. Lys14 and Gcn4 are subject to a major posttranscriptional control, and thus using

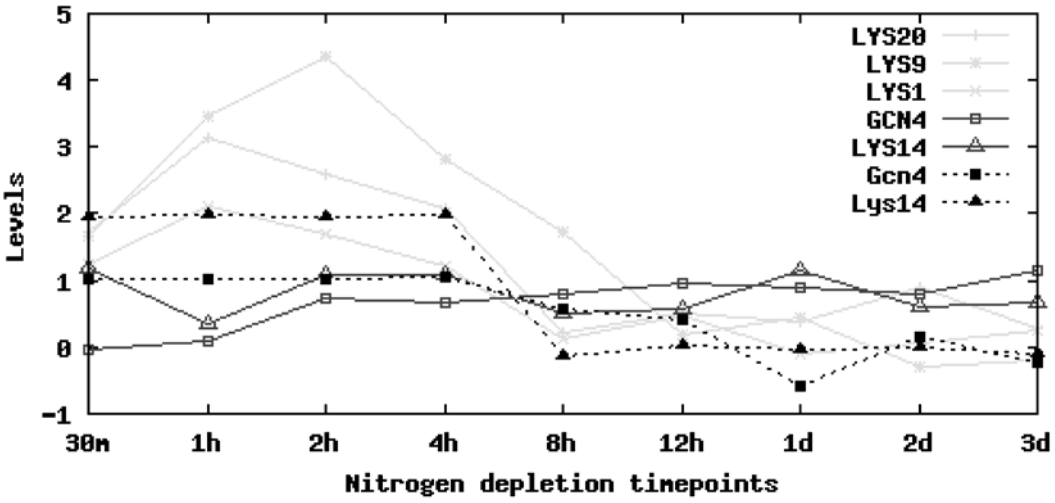


FIG. 10. States of lysine model variables in nitrogen depletion experiments. X axis: time points of the nitrogen depletion experiments of Gasch *et al.* (2000). Y axis: solid lines are measured mRNA levels; broken lines are inferred protein levels. (The mRNA levels are as explained in Gat-Viks *et al.* [2004]; protein levels are computed using a predefined discretization scheme with the arbitrary averages -1, 1, and 3). We plot the observed levels of the mRNAs of the TFs GCN4 and LYS14 in gray empty shapes, and regulatees LYS1, LYS20, and LYS9 in light gray. We also plot expected levels of the proteins Gcn4 and Lys14 as inferred by the model. Note that the mRNA levels of the TFs are roughly constant throughout the experiment, while the model-based inference highlights possible changes in the protein levels, by exploiting the connection between the protein levels and their regulators levels.

the mRNA levels to approximate the protein levels might lead to learning mistakes. We used our learning method to refine the model for the lysine biosynthetic enzymes and were able to learn the known inductive role of each of their regulators. In addition, Lys14 can activate transcription in the absence of Gcn4 activity (see *www.cs.tau.ac.il* for details). The features obtained are similar to the results of Gat-Viks *et al.* (2004), but now we can use the p-values to pinpoint the highly significant features learned.

7. DISCUSSION

In this study, we have introduced a computational framework for the study of biological systems using a combination of prior knowledge on the regulation of system's components with data from diverse high-throughput experiments. We developed a practical approach for exploiting as much of the available information on the system as possible in an integrative fashion. The goals were to systematically test the correctness of prior assumptions on the regulation of the system, by comparing predicted and observed experimental behavior, and to refine our regulation models so that possible model discrepancies are alleviated. Our mathematical formulation offers flexibility that can be used to express knowledge at all levels: In terms of the model, extant knowledge can range from confirmed and quantified regulatory relations to hypotheses and beliefs on poorly characterized parts of the system. In terms of experimental data, these can range from controlled high-throughput experiments, testing the behavior of the system from many possible angles, to high- and low-throughput experiments indicating the activity of only a small fraction of the system's factors. We believe that such a flexible and data-absorbing approach to the learning of models for biological systems is pertinent to making computational tools helpful when addressing concrete biological problems.

In developing the current framework, we have used many simplifications and limiting assumptions, trying to strike the right balance between our wish to construct a faithful description of the biological system and the scarcity of accurate experimental information at very high resolution. In the future, with the anticipated advent of refined understanding of regulatory switches, truly quantitative experiments on more aspects of biological regulation (e.g., protein abundance and states) and measurements at the single cell level, our framework could be extended in several major directions.

In its current form, our model describes the steady state behavior of the system. Biological processes are inherently temporal, but when the sampling rate (the number and time resolution of experiments) is slow relative to the rate of the regulatory mechanisms, the steady state assumption is more practical than other assumptions. We note that different regulatory processes operate on different time scales: In the typical high-throughput experimental sampling rate, the steady state assumption is highly adequate for metabolic pathways and posttranslational regulation and reasonable for transcriptional programs. The models considered in this work included variables of many types, and we validated empirically (using, e.g., cross validation) that the steady state assumption still enables biologically meaningful results with each of them. The model is already capable of handling steady state (or fast) feedback loops, and it will be natural to extend it to handle slower temporal processes in a way analogous to the construction of dynamic Bayesian networks (DBN) (Friedman *et al.*, 1998; Smith *et al.*, 2002) from steady state Bayesian networks. As in DBNs, the algorithms for inference and learning can be naturally generalized from the steady state model to the dynamic model.

Another major simplification we have applied in this work is with the modeling of logical relations using discrete functions (or distributions over discrete functions). We have used this assumption primarily since most of the current prior knowledge on transcriptional switches and other regulatory relations is essentially qualitative. It is clear however that using more biologically justifiable classes of regulation functions (e.g., Tanay and Shamir [2004], Ronen *et al.* [2002], Imoto *et al.* [2004], and Nachman *et al.* [2004]) can help to constrain the learning process toward more significant results.

A final word of caution should be added with respect to topology learning. In the current work, we assumed a fixed topology for the regulatory network. The learning of regulation functions could be performed based on that topology, with reasonable statistical power. In order to enable true topology learning in our framework, much more data or other types of restrictions (e.g., a small repertoire of model variables) would be required. The tools we developed here could be readily applied, however, in settings where structure learning is reasonable (e.g., as in Sachs *et al.* [2005]).

ACKNOWLEDGMENTS

We thank Nir Friedman, Dana Pe'er, and the anonymous referees for helpful comments. I.G.V. was supported by a Colton fellowship. A.T. was supported in part by a scholarship in complexity science from the Yeshaia Horvitz Association. D.R. was supported by a summer student fellowship from the Weizmann Institute of Science. R.S. holds the Raymond and Beverly Sackler Chair for Bioinformatics at Tel Aviv University and was supported in by the Israel Science Foundation (Grant 309/02) and by the EMI-CD project that is funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health," contract number LSHG-CT-2003-503269. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

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. 2003. Computational discovery of gene modules and regulatory networks. *Nature Biotechnol.* 21, 1337–1342.

Beer, M.A., and Tavazoie, S. 2004. Predicting gene expression from sequence. *Cell* 117, 185–198.

Buntine, W.L. 1995. Chain graphs for learning. *Proc. 11th Ann. Conf. on Uncertainty in Artificial Intelligence (UAI '95)*, 46–65.

Chen, K.C. *et al.* 2000. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol. Biol. Cell* 11, 369–391.

Cooper, G. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intell.* 42, 393–405.

Covert, M.W., Knight, E.M., Reed, J.L., Herrgard, M.J., and Palsson, B.O. 2004. Integrating high-throughput and computational data elucidates bacterial networks. *Nature* 429, 92–96.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *J. Comp. Biol.* 7, 601–620.

Friedman, N., Murphy, K., and Russell, S. 1998. Learning the structure of dynamic probabilistic networks. *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, 139–147.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. 2000. Genomic expression programs in the response of yeast to environmental changes. *Mol. Biol. Cell* 11, 4241–4257.

Gat-Viks, I., Tanay, A., Raijman, D., and Shamir, R. 2005. The factor graph network model for biological systems. *Proc. RECOMB 2005*, 31–47.

Gat-Viks, I., Tanay, A., and Shamir, R. 2004. Modeling and analysis of heterogeneous regulation in biological networks. *J. Comp. Biol.* 11, 1034-1049.

Hartemink, A., Gifford, D., Jaakkola, T., and Young, R. 2002. Combining location and expression data for principled discovery of genetic regulatory networks. *Proc. 2002 Pacific Symposium in Biocomputing (PSB '02)*, 437–449.

Hohmann, S. 2002. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol. Mol. Biol. Rev.* 66(2), 300–372.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., and Miyano, S. 2004. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *J. Bioinform. Comp. Biol.* 2, 77–98.

Imoto, S., Kim, S., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., and Miyano, S. 2004. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *J. Bioinform. Comp. Biol.* 1, 231–252.

Jaakkola, T.S. 2001. Tutorial on variational approximation methods, *in* Saad, D., and Opper, M., eds., *Advanced Mean Field Methods—Theory and Practice*, 129–160, MIT Press.

Kschischang, F.R., Frey, B.J., and Loeliger, H. 2001. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory* 47, 498–519.

MacKay, D.J.C. 1998. Introduction to Monte Carlo methods, *in* Jordan, M.I., ed., *Learning in Graphical Models*, 175–204, Kluwer Academic Press, New York.

Nachman, I., Regev, A., and Friedman, N. 2004. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics* 20, 248–256.

O'Rourke, S.M., and Herskowitz, I. 2004. Unique and redundant roles for hog mapk pathway components as revealed by whole-genome expression analysis. *Mol. Biol. Cell* 15(2), 532–542.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, New York.

Proft, M., and Serrano, R. 1999. Repressors and upstream repressing sequences of the stress-regulated ena1 gene in *Saccharomyces cerevisiae*: bzip protein sko1p confers hog-dependent osmotic regulation. *Mol. Biol. Cell* 19, 537–546.

Rep, M., Krantz, M., Thevelein, J.M., and Hohmann, S. 2000. The transcriptional response of *Saccharomyces cerevisiae* to osmotic shock. hot1p and msn2p/msn4p are required for the induction of subsets of high osmolarity glycerol pathway-dependent genes. *J. Biol. Chem.* 275, 8290–8300.

Rep, M., Reiser, V., Holzmüller, U., Thevelein, J.M., Hohmann, S., Ammerer, G., and Ruis, H. 1999. Osmotic stress-induced gene expression in *Saccharomyces cerevisiae* requires msn1p and the novel nuclear factor hot1p. *Mol. Cell. Biol.* 19, 5474–5485.

Ronen, M., Rosenberg, R., Shraiman, B., and Alon, U. 2002. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci. USA* 99, 10555–10560.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., and Nolan, G.P. 2005. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 308(5721), 523–529.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. 2003. Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genet.* 34(2), 166–176.

Smith, V.A., Jarvis, E.D., and Hartemink, A.J. 2002. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics* 18, 216–224.

Tanay, A., and Shamir, R. 2001. Computational expansion of genetic networks. *Bioinformatics* 17, S270–S278.

Tanay, A., and Shamir, R. 2004. Modeling transcription programs: Inferring binding site activity and dose-response model optimization. *J. Comp. Biol.* 11, 357–375.

Yeang, C.H., Ideker, T., and Jaakkola, T. 2004. Physical network models. *J. Comp. Biol.* 11(2–3), 243–262.

Yedidia, J.S., Freeman, W.T., and Weiss, Y. 2004. Understanding belief propagation and its generalizations, *in* Lakemeyer, G., and Nebel, B., eds., *Exploring Artificial Intelligence in the New Millennium*, 239–269, Morgan Kaufmann, New York.

Yedidia, J.S., Freeman, W.T., and Weiss, Y. 2004. Constructing free energy approximations and generalized belief propagation algorithms. Technical report TR-2004-040, Mitsubishi Electric Research Laboratories.

Address correspondence to:
Irit Gat-Viks
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: iritg@post.tau.ac.il

Methods

Refinement and expansion of signaling pathways: The osmotic response network in yeast

Irit Gat-Viks[1] and Ron Shamir

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel

The analysis of large-scale genome-wide experiments carries the promise of dramatically broadening our understanding on biological networks. The challenge of systematic integration of experimental results with established biological knowledge on a pathway is still unanswered. Here we present a methodology that attempts to answer this challenge when investigating signaling pathways. We formalize existing qualitative knowledge as a probabilistic model that depicts known interactions between molecules (genes, proteins, etc.) as a network and known regulatory relations as logics. We present algorithms that analyze experimental results (e.g., transcription profiles) vis-à-vis the model and propose improvements to the model based on the fit to the experimental data. These algorithms refine the relations between model components, as well as expand the model to include new components that are regulated by components of the original network. Using our methodology, we have modeled together the knowledge on four established signaling pathways related to osmotic shock response in *Saccharomyces cerevisiae*. Using over 100 published transcription profiles, our refinement methodology revealed three cross talks in the network. The expansion procedure identified with high confidence large groups of genes that are coregulated by transcription factors from the original network via a common logic. The results reveal a novel delicate repressive effect of the HOG pathway on many transcriptional target genes and suggest an unexpected alternative functional mode of the MAP kinase Hog1. These results demonstrate that, by integrated analysis of data and of well-defined knowledge, one can generate concrete biological hypotheses about signaling cascades and their downstream regulatory programs.

[Supplemental material is available online at www.genome.org.]

Genome-wide expression profiles (Gasch et al. 2000; Hughes et al. 2000) have paved the way to systems biology approaches that aim to elucidate system architecture by large-scale data analysis. A variety of sophisticated computational methods have been developed toward this goal (Eisen et al. 1998; Ihmels et al. 2002; Beer and Tavazoie 2004; Friedman 2004). An essential and important part of these analyses is the biological interpretation of the computational results based on knowledge available in the literature. The common practice is to first perform the computational analysis and then to explain the results using prior knowledge (Tavazoie et al. 1999). However, several studies have shown the advantage of integrating the existing knowledge as part of the analysis (Ideker et al. 2001; Gardner et al. 2003; Covert et al. 2004; Gat-Viks et al. 2004). In this study we propose a new method that aims to achieve a better understanding of a signaling pathway by integrated analysis of genome-wide datasets and prior knowledge, in a way that improves that knowledge systematically. The method suggests new hypotheses which can be validated by additional focused experiments.

We formalize the current information on the studied biological system in a mathematical model. Cellular signaling networks are characterized by signal transduction pathways that are triggered by environmental stimulation and control the cellular response. For such biological systems, a large body of qualitative knowledge is available today, both on the structural and on the logical relations between the components. In many cases, the information is still informal and thus not amenable to mathematical manipulation. For example, many transcription factors have been established as activators or repressors, but their stoichiometric coefficients are unknown. To properly formalize such qualitative knowledge, we use Bayesian networks, a probabilistic framework for modeling complex systems such as signaling cascades (Sachs et al. 2002; Friedman 2004). Our model formalizes the current knowledge about the structure ("topology") of the network, i.e., which system components interact, and its logic, which dictates the level of each component based on the level of its upstream effectors (Gat-Viks et al. 2006). The topology tells "which component acts on which other components" and the logic tells "how that action takes place."

The model predicts the levels of the system's variables (genes, proteins, etc.) under each condition and is improved systematically in a process that seeks structural and logical changes that increase the fit between predicted and observed variable levels. In particular, we propose two methods for model improvement (Fig. 1): The first refines the model by adding interactions and modifying logics, without adding variables. The second expands the model to include additional variables beyond the original model. We focus on the identification of regulatory modules, i.e., sets of coregulated genes that are regulated by the same model components via a common logic. In the standard clustering approach, after identifying a group of coregulated genes, the regulating transcription factors are revealed by overrepresentation of their DNA binding motifs, or by enrichment in chip-ChIP data (Beer and Tavazoie 2004). In contrast, using our methodology, the newly discovered modules are added to the model, and thus their regulators and the logic of their regulation are determined as part of the analysis. Consequently, the expression of the modules is directly explained by the model.

[1]Corresponding author.
E-mail iritg@tau.ac.il; fax 972-3-6405384.

Article published online before print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.5750507.

17:358–367 ©2007 by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/07; www.genome.org

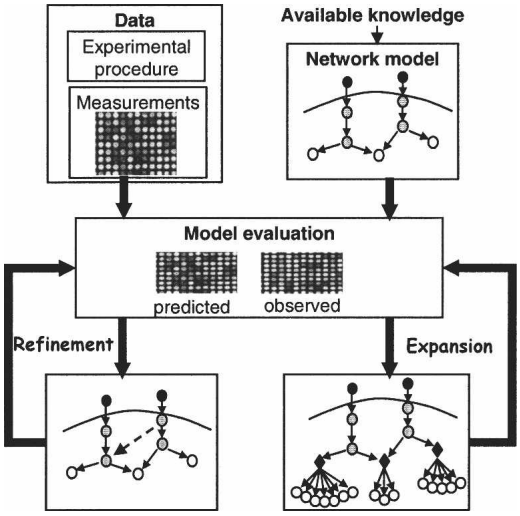


Figure 1. Overview of the model improvement methodology. Model formalization: The current qualitative knowledge on the studied biological system is formalized as a Bayesian network (*top right*; see also Fig. 2). The illustrated model contains several molecular types: environmental stimulations (dark gray), signaling proteins and transcription factors (light gray), and mRNAs (white). The refinement and expansion procedures take as input the network model and high throughput measurements on network's components (*top left*), and search systematically for model improvements that maximize a probabilistic improvement score. The score measures the increase of fit between the model predictions and the observed data. The model refinement procedure (*middle left*) seeks structural and logical changes in existing model components, which attain the best score. Structural refinements are marked by dashed connections. The model expansion procedure (*middle right*) assigns systematically new target genes to regulatory modules, based on their fit to the predicted expression of the module. In the illustration, three regulatory modules were formed. They contain known and novel target genes (white circles). All genes in the same module share the same logic (black diamonds).

We have chosen to apply our methodology in the analysis of the cellular response of *Saccharomyces cerevisiae* to hyper-osmotic and calcium stresses. This response is mediated by a signaling network that involves the PKA signaling pathway, the HOG and mating/pseudohyphal growth MAPK cascades, and the calcineurin pathway. Based on 106 transcription profiles (Gasch et al. 2000; Harris et al. 2001; Yoshimoto et al. 2002; O'Rourke and Herskowitz 2004), the refinement procedure suggests three missing cross-talk connections in the network, which all have independent support in the literature. The expansion procedure was applied to six known regulatory modules and 78 putative sets of regulators and yielded 10 statistically significant modules. We discover both HOG pathway-dependent induced and repressed novel modules, and show that these modules are distinct from the known HOG pathway-dependent response. Remarkably, our analysis indicates that Hog1 MAP kinase acts in several distinct functional modes. The expanded network contains many transcriptional regulatory feedback and feedforward loops. This rich circuitry is probably part of the osmotic adaptation and provides rapid and transient response to osmotic changes.

Several features distinguish our computational methodology from extant network reconstruction methods. Recently, a few advanced methods sought to improve system models systematically, both for quantitative metabolic networks (Klipp et al. 2005; Herrgard et al. 2006) and for physical interaction networks (Calvano et al. 2005; Yeang et al. 2005). Our approach differs in that it uses informal qualitative knowledge, including regulatory logics, which is crucial for modeling of the activation and down-regulation of signaling cascades. Bayesian networks were used for de novo reconstruction of system models (Friedman 2004). In contrast, here the Bayesian network represents the existing well-characterized system model, and the analysis seeks its improvement. In addition, we use a discriminative improvement score, rather than a classical Bayesian score, in order to identify significant and specific model changes. Concerning modules identification, extant methods approximate the regulator's protein activity by its mRNA expression (Bar-Joseph et al. 2003; Segal et al. 2003; Tamada et al. 2003). A key advantage of our methodology is that we use the model to predict the activity of the regulators, and then use these levels to identify the modules. Since the transcription factor activity levels are more directly related to their targets' expression, better module identification is possible.

Overall, the results show that, by formalizing the qualitative knowledge available and analyzing the system model jointly with relevant large-scale data, it is possible to extend the current understanding on biological systems and to analyze regulatory mechanisms in a new level of detail.

Results

We selected for our analysis 106 gene expression profiles from four large-scale microarray studies in yeast (Gasch et al. 2000; Harris et al. 2001; Yoshimoto et al. 2002; O'Rourke and Herskowitz 2004). The profiles measure the yeast response to osmotic and calcium stresses and the effect of genetic perturbations in the osmotic response pathways. Originally, these studies applied clustering algorithms on the data. The following results show that, by integrated analysis of the data and the model, we find regulatory relations and mechanisms that could not be revealed using the data alone.

The computational approach

We formalize the biological knowledge in a Bayesian network model (Gat-Viks et al. 2006), which represents dependencies among interacting components. The components, or *variables*, are mRNAs, proteins, external inputs, etc. The model provides a *structure* and a *logic* for each variable. The structure (or topology) is represented by a graph diagram, where the nodes represent the variables, and arcs represent influence among variables (e.g., transcription factor binding to a gene promoter, phosphorylation by a kinase, etc.). For each graph node, the nodes that have arcs directed into it are its *regulators,* or its *regulatory unit.* Each variable can be in one of several discrete *states*, indicating, for example, the activity of a protein variable, or the expression level of a mRNA variable. In the logic component of the model, a variable's state is determined by the combination of states of its regulators according to its specified discrete function, which might represent a complex relationship among multiple regulators. The logic is formulated probabilistically in order to allow for uncertainty about the available biological knowledge (Fig. 2A).

In order to allow formulation of the available qualitative knowledge, we have chosen to model the logics as discrete functions using discrete states. However, the actual cellular concentrations are continuous levels, and hence our model must transform continuous levels into discrete logical states. The *observed level* (or *observation*) is the result of a measurement in a biological experiment, e.g., the measured concentration of mRNAs or a metabolite, or the measured phosphorylation of a protein which

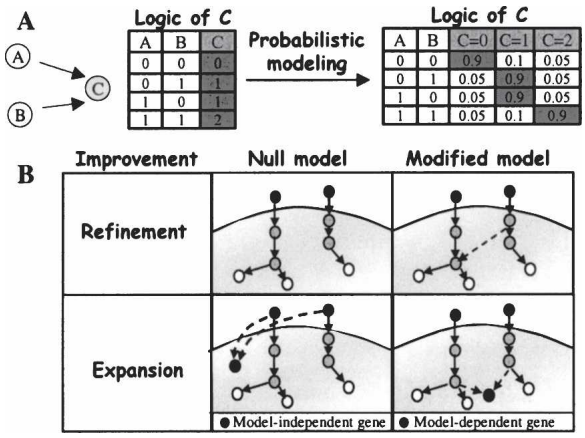


Figure 2. The computational approach. (*A*) Modeling the current knowledge. Nodes represent the variables of the model and arcs are known regulatory relations. Here, the state of variable C depends on the states of its regulators A and B according to a specific logic. In the combinatorial logic of C (*left*), the state of C is 1 if, and only if, at least one of its regulators has state 1. In the probabilistic modeling (*right*), each possible state of C is assigned a probability depending on our confidence in the current biological knowledge (here, 90% confidence). (*B*) Improving the model. The model refinement and expansion procedures look for model changes that improve the model significantly. The improvement score compares between the fit of a possible modified model and that of the null (original) model. The plots are a schematic representation of these two models in cases of refinement (*top*) and expansion (*bottom*). In expansion, when adding a new gene, the null model assumes that the gene expression can be explained sufficiently by the environmental stimulation. The alternative hypothesis is a model-dependent gene, i.e., the gene is regulated by our signaling network. We expand the model only if the improvement score is significant, i.e., the signaling network explains the expression much better than the environmental stimulation only.

indicates its activity. The *predicted level* is the probabilistic expectation of the variable given the model and the experimental procedure applied (i.e., the genetic perturbations and the environmental stimulation performed in the experiment). Hence, the predicted levels of protein activities (*predicted activities*) constitute additional information that is not available from microarray experiments. The predicted levels of mRNA variables (*predicted expression*) are compared to the observed expression, and reveal important information on the quality of the model. In particular, points of disagreement between observed and predicted expression levels indicate where our understanding of the biological system is lacking. Mathematically, the quality of the model is evaluated by a *Bayesian score*, which measures the closeness of the observations to the predicted levels (see Methods).

Naively, the model can be improved by searching in the space of all possible model improvements (i.e., either refinements or expansions) for the model with the best Bayesian score. However, in order to propose only trustable hypotheses, we introduce here a new *improvement score*, which measures the difference between the Bayesian scores of the modified and the original model. Hence, we seek model improvements with significantly high improvement scores. In the case of model refinement, the improvement score compares the Bayesian score before and after introducing the logical or structural changes (Fig. 2B). In the more complicated case of model expansion, among all genes that respond to the environmental changes, we wish to identify specifically the *model-dependent genes*, which are affected by model components. We wish to exclude other responding genes (*model-independent genes*), such as ribosomal proteins, which respond to the environmental stimuli, but probably independently of our model and through another signaling pathway. Both types respond to the environmental changes, but only the model-dependent responding genes are influenced by genetic perturbations in model components. Hence, the expansion improvement score compares the scores of adding a gene in a model-dependent and in a model-independent fashion (Fig. 2B). A gene with a significant improvement score is considered a model-dependent gene and is assigned to the module (i.e., regulatory unit and logic) that obtained the highest improvement score (see Methods).

The osmotic response network model in yeast

Building on literature reports, we modeled the response of yeast cells to calcium and hyper-osmotic stresses. The model formalizes the HOG, mating/pseudohyphae growth, calcineurin, and the PKA signaling pathways. The signaling cascades act together to affect the activity of many regulators (Hog1, Sko1, Msn1, Hot1, Msn2/4, Crz1, Ste12) that govern the complex expression of target genes by diverse combinatorial logics. For each pathway, our model includes the environmental stresses, the signaling cascades, the transcription factors, and their known targets (Fig. 3). Each variable has three to five possible states. Supplement A catalogs all variables, connections, and logics in the model, along with their source in the literature. All the literature sources used for the modeling do not rely, directly or indirectly, on the 106 profiles that we use here. Note that the mating and pseudohyphal growth pathways are modeled together. Since they share most of their components (up to the Kss1/Fus3 MAPKs and their upstream activators; O'Rourke and Herskowitz 1998), and our dataset does not include any experiment that can distinguish between them, a separate modeling of the pathways will not improve our results. In practice, our joint modeling of mating/

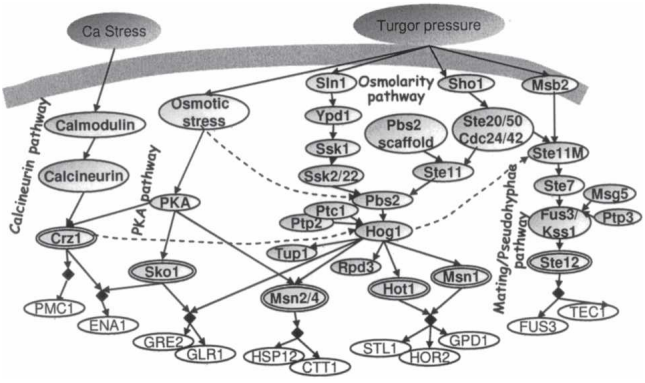


Figure 3. A model of the yeast response to osmotic and calcium stress. The model contains (*left* to *right*) the calcineurin pathway, PKA signaling pathway, the HOG MAPK pathway, including its Sln1–Ssk1 and Sho1–Ste11 upstream branches, and the mating/pseudohyphal growth pathways. The network, constructed based on literature reports, contains environmental conditions (dark gray), signaling components (light gray ovals), transcription factors (double ovals) and their transcriptional targets (white ovals). Targets sharing the same regulatory logics (i.e., in the same module) are indicated by black diamonds. Arrows are well-established relations (solid lines) or relations predicted by the refinement procedure (dashed lines). The logic by which each component is governed by its regulators is described in Supplement A. The dual role of the MAPKKK Ste11 in the HOG and mating pathways is formalized by refining two different model variables called Ste11 and Ste11^{M}, respectively.

pseudohyphae pathways reduces model size and thus increases efficiency and accuracy.

Network refinements

Given the dataset of 106 transcription profiles and the osmotic response model, the refinement procedure looks for structure and logic modifications with high improvement scores. Three new connections providing the most significant improvement (marked as dashed arcs in Fig. 3) indicate cross talk in the model. The three predicted connections are not well-established, and thus were not included in the original model, but each has an independent support in the literature (Supplemental Table S1). First, the model predicts a down-regulation of the HOG pathway by the calcineurin pathway (Crz1⊣Pbs2/Hog1, improvement score P-value < 0.0005, see Methods). Indeed, Shitamukai et al. (2004) support this claim by showing that calcium ions induce Hog1 hyperphosphorylation in *crz*1 mutants. Calcium ions activate Crz1 through the calcineurin pathway and activate Hog1 through the HOG pathway. Crz1 down-regulates Hog1 and thus there is hyperphosphorylation of Hog1 in a strain lacking Crz1. Second, Hog1 prevents osmolarity-induced activation of the mating/pseudohyphae pathway. The predicted inhibitory connection is directed from Hog1 to the mating MAPKKK Ste11 or to its downstream mating components, but not to the osmosensor Sho1 (P-value < 0.005). Indeed, the data show strong inhibition of the mating/pseudohyphae targets in *sho1* mutant (Supplemental Fig. S1A), and thus the refinement procedure could not predict that the inhibition is directed to Sho1, but only to its downstream components. O'Rourke and Herskowitz (1998) suggested this cross talk based on measurements of morphological changes and mating phenotypes.

Third, an alternative mechanism is proposed for HOG pathway activation in severe osmotic shock. Significant improvements (P-value < 0.0005) were obtained for the connections: Osmotic Stress → Ssk2/22 and Osmotic Stress → Pbs2. The HOG pathway is still active in *ssk1sho1*, *ssk1ste11* mutants, but not in *pbs2* or *hog1* mutants (Supplemental Fig. S1B), and thus a third input to Ssk2/22 or Pbs2 was added by the refinement procedure. Van Wuytswinkel et al. (2000) provide an independent support for the existence of such additional input to Pbs2. Note that O'Rourke and Herskowitz (2004) already observed this effect in their dataset, but here we succeed to identify it automatically.

The model expansion process

A regulatory module is a set of genes that are regulated by the same regulatory unit via the same logic. To expand the network model, we focused on identifying such modules whose regulatory units are part of the original model. In principle, the space of possible modules is huge: All subsets of variables in the model may participate in a regulatory unit with any possible logic. In practice, we tested putative regulatory units of one or two variables, including the six known units depicted in Figure 3. Altogether, the number of tested units was 78, among them 72 putative units, each with up to 3^9 = 19,683 possible discrete logics, and six known units, each with its known logic (see details in Supplemental Fig. S2). All 5700 measured yeast genes were considered as possible targets, each with three possible states.

For each target gene, the expansion procedure searches heuristically for the unit and logic that best predict its expression as a function of the predicted activity of the regulators. The predicted activities represent the post-transcriptional effects that are formalized in our model, such as the regulator's phosphorylation (and hence activation) by the MAPK Hog1. An alternative approach is to approximate activity with expression levels (Friedman et al. 2000; Tamada et al. 2003), but this approach cannot handle the major post-translational regulation events in the osmotic signaling cascade (Supplemental Fig. S3).

As described above, in order to avoid inclusion of nonspecific targets, the expansion procedure computes the improvement score and thus discriminates between model-dependent responding genes and model-independent responding genes (see Methods and Fig. 2B). According to this analysis, while about 71% of the yeast genes respond to the osmotic stress, only 15% are specifically dependent on the model. On the other hand, the fact that a fifth of the stress response is characterized as model-dependent highlights the important role of the osmotic-specific stress mechanisms in the general cellular machinery of response to stress.

Since small modules could have been generated at random given the large space of regulatory units and logics searched, we focused further analysis on novel modules containing at least 20 genes, and known modules of at least 10 genes. Five novel modules and five known ones passed this filter. When performing expansion using randomly shuffled condition labels (experimental procedures), no module with more than three genes was found (Supplemental Fig. S5), indicating that it is unlikely to obtain our large modules at random.

Transcriptional modules discovered

The known regulatory units of Msn2/4, Ste12, Hot1/Msn1, Crz1, and Sko1 attained modules containing 52, 32, 15, 13, and 12 genes, respectively (Fig. 4; Supplement C). The Crz1–Sko1 unit was assigned only its known *ENA1* target gene. We discovered three novel modules regulated by Hog1 with different logics (referred to as Hog1A, B, and C), one module controlled by both Hog1 and calmodulin (called Hog1/Ca), and one module regulated by Ssk2/22 or Ssk1, called Ssk2/22 (Supplement C).

The predicted regulatory units do not necessarily control their target genes directly. For example, the Msn2/4-module contains *YAP4*, (currently known as *CIN5*), *GCY1*, and *DCS2*, but, actually, Msn2 regulates the *YAP4* gene, which encodes a transcription factor; the up-regulation leads to increased activity of Yap4, which in turn up-regulates transcription of *GCY1* and *DCS2* (Nevitt et al. 2004). Calmodulin and Ssk2/22 probably affect their targets indirectly, since they are cytoplasmatic kinases and have no DNA binding domain. The prediction that their regulatory effect does not involve downstream elements in the model has some support in the literature (Ohya et al. 1991; Yuzyuk et al. 2002).

A key advantage of our methodology is that the activity of the modules can be predicted by the model and compared with the observed levels. Cases of disagreement between the predicted and observed levels are of particular interest, since they highlight spots of incomplete understanding in the biological system. For example, the Ste12 module shows inconsistency in the case of *ssk1sho1* mutants exposed to 0.5 M KCl and the *ssk1* mutants exposed to 0.125 M KCl (marked in Fig. 4; an extended version of this module appears in Supplement C). An increase in transcription is observed, in contrast to the predicted reduction. The inaccurate modeling is probably due to incomplete understanding of the inhibitory effect of Hog1 on the mating/pseudohyphal growth pathway.

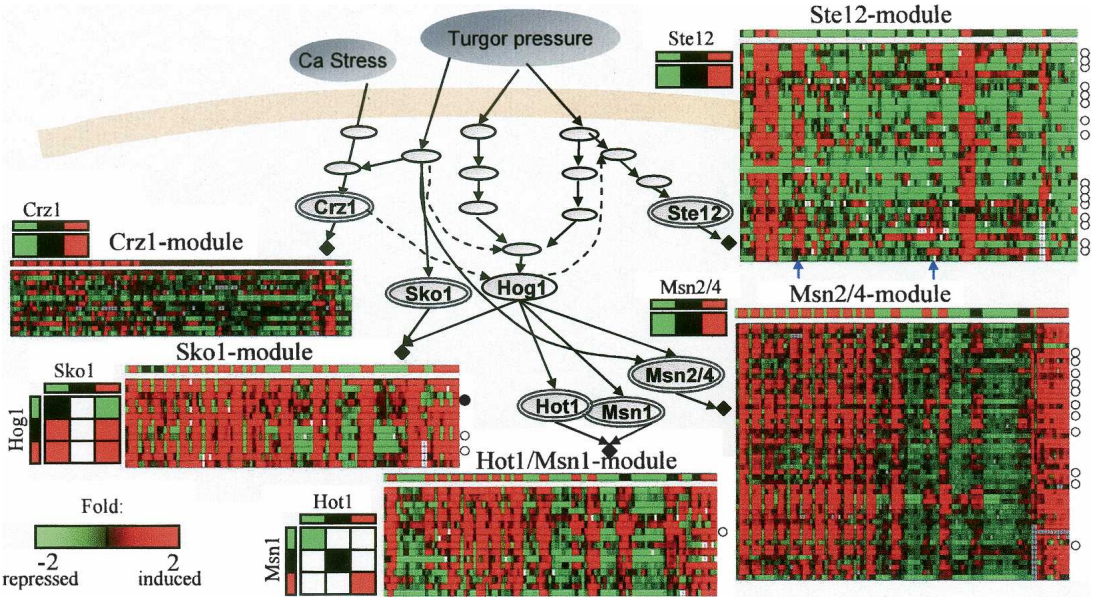


Figure 4. Expansion of the osmotic network model. The expansion algorithm assigns known and novel target genes to known modules (black diamonds). Each module is represented by a matrix showing the expression of its target genes (rows) across the 106 conditions (columns). Known target genes that were assigned to their module correctly/incorrectly are marked with white/black circles to the *right* of the corresponding row (known targets were excluded from the model before expansion, to allow validation and to avoid circularity). The predicted expression levels in each condition appear as a separate row *above* the matrix. The logic of each module, obtained by the refinement procedure, appears *near* the matrix. We show in color only logic entries with significant improvement score. In general, there is high agreement between model predictions and observed levels. The few cases of disagreement (e.g., columns marked by blue arrows in the Ste12 module) highlight our incomplete understanding (and hence modeling) of the biological system. The full details on each module appear in Supplement C, including lists of correct/incorrect target genes, and their sources in the literature.

Transcriptional modules evaluation

A unique feature of our methodology is that a module and its regulators are identified together in the same process. In order to evaluate the methodology, we excluded all known transcriptional targets from the model and then constructed the modules. We then tested the accuracy of assigning known targets to modules. An extended collection of 126 known targets and their literature sources is available in Supplement C. Among them, 37 genes were assigned to modules, and 17 additional genes were assigned to very small modules which were filtered from in our analysis. Out of the 37 genes assigned to modules, 30 genes were assigned correctly to their known regulators, and one gene was assigned incorrectly (marked in Fig. 4). Six additional Msn2/4 targets were assigned to the Hog1A novel module, which is also hypothesized to be regulated by Msn2/4 (see below). Hence, we obtain 97% specificity (correct/assigned = 36/37; see Supplemental Table S2). To get such high specificity, we pay the cost of low (29%) sensitivity (correct/known = 36/126).

In another evaluation of the predicted modules and their regulators, we tested each module for enrichment in transcription factor (TF) binding using TF-DNA binding profiles (Harbison et al. 2004). For each TF whose binding profile in relevant conditions is available, the enrichment test supports the predicted regulatory unit (Supplemental Fig. S6A): The Ste12 module is bound by Ste12, Dig1, Mcm1, and Tec1 in mating/PH growth induction (pheromone and Butanol treatment); the Msn2/4 module is bound by the Msn2/4 in stress conditions (acidic and H_2O_2 treatment); and the Sko1 module is bound by Sko1 in YPD medium. Indeed, Sko1-dependent repression is constitutively active (bound) under normal conditions and derepressed under osmotic shock. In addition, for the modules of Ste12 and Msn2/4, sequence analysis shows that the known TF binding site motifs are highly enriched in the promoters of the genes in the predicted module (Supplement D).

To validate the biological significance of the predicted gene sets, we tested the functional coherence and separation of gene sets. We used 87 gene expression profiles of 10 stress conditions from Gasch et al. (2000) that were not included in the set of 106 profiles used for constructing the modules (stationary phase, heat shock, Diamide, Menadione, H_2O_2, amino acid starvation, nitrogen depletion, hypo-osmotic shock, DTT, and various carbon sources). We found significant coregulation of the genes in each module and significant separation between modules (Supplemental Fig. S6B,C). The module predicted to be regulated by Msn2/4 shows strong coregulated response in all stress conditions, in agreement with the known general stress functionality of the Msn2/4 transcription factors.

Separating gene sets that differ only in a few experiments using standard clustering algorithms is a hard task, since the minor expression differences might be the result of noise. A unique feature of our approach is the ability to separate genes using both data and prior model, rather than data only. Hence, if the model can predict two modules with slight differences, these differences become significant, and the targets will be partitioned into two modules. For example, the targets of Hog1B module and Ssk2/22 module were separated by the model, even though they are very similar according to our data (Supplemental Fig. S6D). The separation is corroborated using independent data of heat shock stress (Gasch et al. 2000), in which the expression patterns of these two gene sets are significantly different (KS-test P-value $< 10^{-3}$; Supplemental Fig. S6E). Another

example for separation of two similar Msn2/4 modules is given below.

The transcription factors Msn2/4 regulate two distinct modules

In our analysis, we identified the known Msn2/4 module (Fig. 5A). In addition, several indications suggest that Hog1A, one of the novel modules (Fig. 5B), is also regulated through Msn2/4. First, Hog1A is enriched in Msn2/4 targets: Among 24 module genes known to be Msn2/4 targets (based on expression experiments in Msn2/4 knockout mutants from Rep et al. 2000), 11 are in the Msn2/4 module, and seven are targets of the Hog1A module (Fig. 5A,B, hyper geometric enrichment P-value $< 10^{-17}$, 10^{-12}, respectively), and the rest were assigned to various other logics. Second, a significant enrichment in binding of Msn4 to the promoters of Hog1A module genes was observed in ChIP experiments (Harbison et al. 2004) ($P < 10^{-7}$; Supplemental Fig. S6A). Third, the Hog1A module is highly expressed in strains overexpressing Msn4 (Gasch et al. 2000) (two right columns in Fig. 5A; KS-test P-value $< 10^{-12}$). Fourth, Hog1A exhibits highly significant response in all stress conditions (Supplemental Fig. S6B), in agreement with the central role of Msn2/4 in general stress response. Finally, the Msn2/4 STRE binding motif was highly enriched in the Hog1A module (P-value $< 10^{-5}$; Fig. 5D).

To provide additional evidence that the two transcriptional modules are distinct, we performed promoter sequence analysis. Remarkably, a new motif was discovered to be highly enriched only in the novel module (KMCTWGAA, enrichment P-value $< 10^{-14}$) and it may contribute to the unique behavior of the module (Fig. 5C,D). This novel motif exhibits a very strong bias in orientation and distance from the transcription start site of the regulated genes (hyper geometric P-value $< 2 \times 10^{-4}$).

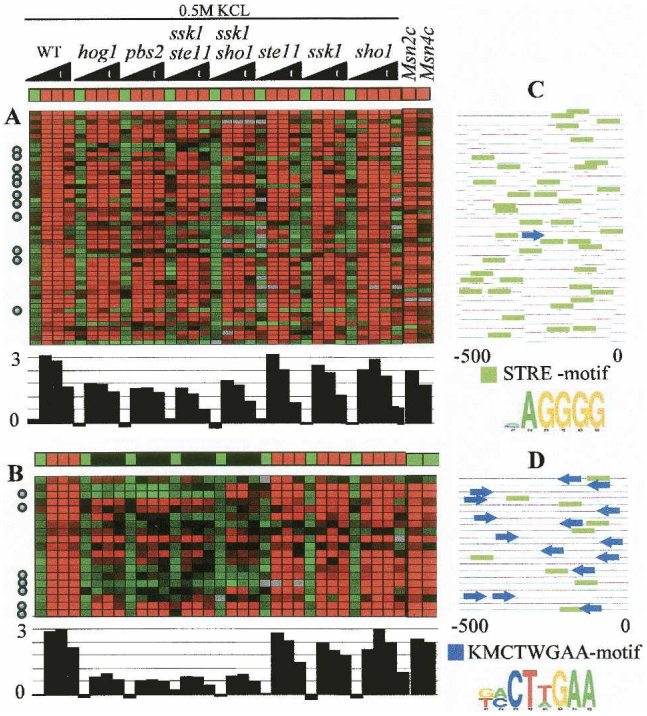


Figure 5. Expression profiles of two modules associated with Msn2/4. (*A*) The known Msn2/4 module. (*B*) The novel Hog1A module. The conditions are time series measurements in response to 0.5 M KCl osmotic shock. *Below* the predicted expression vector and the observed expression matrix (the same presentation as in Fig. 4), the average fold induction of the module is shown. Both modules are hypothesized to be regulated by Msn2/4 and include many known Msn2/4 targets (marked with circles). However, their expression patterns are clearly distinct: The Hog1A module depends much more strongly on the presence of Hog1 in severe osmotic shock. In wild type (WT), the expression level in both modules is ~3, but in *hog1*, *pbs2*, *ssk1ste11*, and *ssk1sho1* the expression levels differ significantly: ~0.5 in Hog1A and ~2 in Msn2/4 module (KS-test P-value $< 10^{-4}$). The two *rightmost* columns in *A* and *B* show the expression level of the modules in Msn2 and Msn4 overexpression mutants. Although the predicted expression in these conditions is low in the Hog1A module, the observed level in both modules is high, indicating that both modules are regulated by Msn2/4. (*C*,*D*) Promoter analysis. Each line represents the 500-bp sequence upstream of the transcription start site for the gene in that row. Green boxes represent occurrences of the STRE motif (a known Msn2/4 binding site); blue arrows represent the new motif KMCTWGAA discovered in this analysis. This motif exhibits a non-uniform distribution along the promoter in terms of location and orientation. The novel motif supports the separation of the Msn2/4 targets into two distinct modules.

HOG pathway-dependent repression of genes

It was previously demonstrated that Hog1-dependent genes are either induced or repressed in *hog1* mutants. The prevalent view in the literature is that the genes induced by *hog1* mutants are associated with pheromone response and pseudohyphal growth (O'Rourke and Herskowitz 2004). Indeed, among nineteen genes that are specifically up-regulated in *hog1* mutants (Rep et al. 2000), all 11 genes with high score (improvement score > 0.05) were assigned to the module of the mating/pseudohyphae TF Ste12. Surprisingly, our results revealed four additional modules that increase specifically in the *hog1* mutant (Fig. 6A; Supplement C). In contrast with the Ste12 targets (Fig. 6B), the novel modules respond neither to pheromone nor to perturbation in the mating/pseudohyphal growth pathway (Supplemental Fig. S7) and are not bound by the TFs Ste12, Tec1, or Dig1/2 (Supplemental Fig. S6A). Taken together, these observations suggest that Hog1 plays an additional role in inhibiting expression that is not related to the cross talk between the HOG and mating/pseudohyphae pathways.

Multiple functional modes of Hog1

The refinement procedure suggested the existence of an alternative third mechanism that activates the HOG pathway in severe osmotic stress, in addition to the two known upstream branches of the pathway (Sho1–Ste11 and Sln1–Ssk1; Fig. 3). This refinement was suggested since the transcription of some of the classical HOG pathway targets (regulated by Hot1, Msn1, and Sko1) does not depend on the two upstream branches in 0.5 M KCl (Fig. 6B). However, the transcription level of the known Msn2/4 targets does depend on the two branches (Fig. 6B). This suggests that Hog1 has two different activity modes, and that one of the modes is only functional while interacting with Msn2/4. To test this prediction computationally, we added to the model, in addition to a Hog1 variable that is controlled by three inputs (the two HOG pathway upstream branches, and a third uncharacterized input), an additional variable called $Hog1^{(2)}$, which is controlled solely by the two HOG pathway upstream branches (Supplemental Fig. S2). We applied the module identification process on this extended model.

Remarkably, although the classical HOG pathway targets seem to be activated by a third input, four novel modules (Hog1A, Hog1B, Hog1C, and Hog1/Ca) are predicted to be regu-

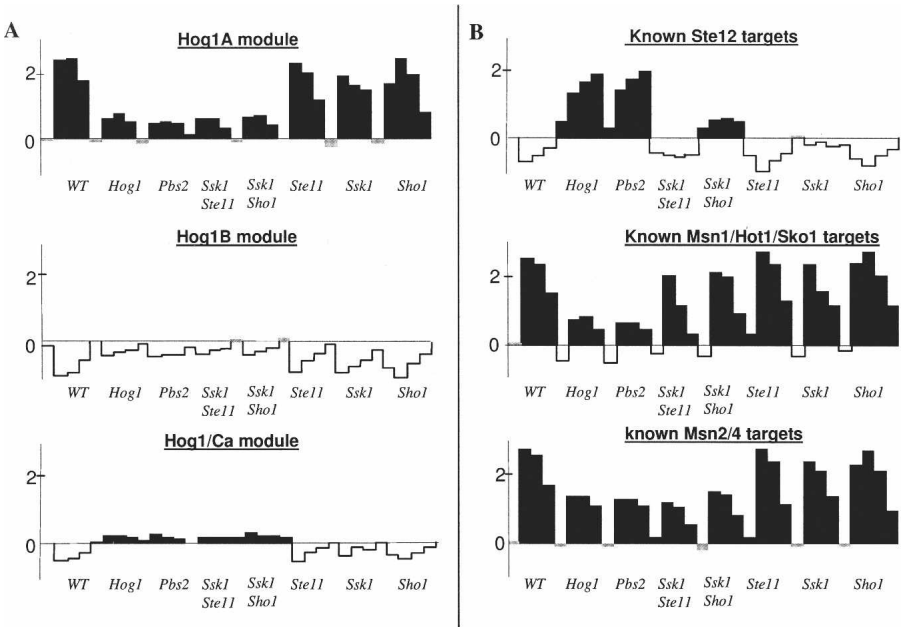


Figure 6. HOG pathway-dependent repression of genes, and multiple functional modes of Hog1. Each plot shows the average fold induction (in log2 scale) of novel gene modules (*A*) or known targets of TFs (*B*) in wild type (WT) and seven HOG pathway mutants exposed to 0.5 M KCl. Black/white coloring indicates average fold induction above/below 0.1. (*A*) The novel modules Hog1A, Hog1B, and Hog1/Ca (the Hog1C and Ssk2/22 modules [data not shown] are similar to Hog1B in this view). (*B*) The known target genes of Ste12 (*KSS1*, *TEC1*, *FUS1*, *FUS3*, *MSG5*, *KAR4*, *CLN1*, *PGU1*), Hog1 (*HOR2*, *GRE2*, *STL1*, *ENA1*, *GLR1*, *GPD1*, *HAL1*, *CHA1*, *AHP1*, *YGR043C*, *YGR052W* (reserved name *FMP48*), *YML131W*; Hohmann 2002), and *Msn2/4* (Rep et al. 2000). Expression of the novel modules Hog1B and Hog1/Ca (*A*, *middle* and *bottom*) increases in the absence of Hog1. Although the whole Hog1-dependent inhibition response is known to be regulated by Ste12, one can clearly see that these novel modules differ significantly from the Ste12 targets (*B*, *top*), indicating existence of Hog1-dependent in spite of Ste12-independent inhibition. The known Hog1/Msn1/Sko1 and Msn2/4 targets (*B*, *middle* and *bottom*) have distinct expression pattern (KS-test *P*-value $< 10^{-5}$): The Msn1/Hot1/Sko1 targets have higher expression in the *ssk1ste11* and *ssk1sho1* mutants compared to *hog1* and *pbs2* mutants, indicating that Hog1 can be activated also by a third additional input. In contrast, the Msn2/4 targets have a similar expression pattern in all four of these mutants, indicating that Hog1 is dependent on the two upstream branches of the HOG pathway. Surprisingly, the novel modules' expression pattern (*A*) also suggests dependency on the two HOG branches. One can clearly see that two of these modules (Hog1B and Hog1/Ca) differ significantly from the known Msn2/4 targets (the distinction between Msn2/4 and the third module Hog1A is discussed in Fig. 5). Taken together, this suggests that Hog1 has two distinct functional modes that involve a different combination of transcription factors. An extended version of the novel modules appears in Supplement C.

lated by Hog1$^{(2)}$ and indeed seem to be dependent on the two upstream branches, similarly to Msn2/4 (Fig. 6; Supplement C). Several indications suggest that one of these modules, Hog1A, is actually regulated through Msn2/4 (as detailed above; Fig. 5). But surprisingly, the Hog1B, Hog1C, and Hog1/Ca modules are not enriched according to any of these criteria, and thus it seems that their regulation does not involve Msn2/4. Therefore, there is a strong indication that Hog1 has multiple functional modes that probably go beyond its functionality in particular combinatorial regulation with Msn2/4. Supporting this new hypothesis, some of these functional modes have opposite effects (there are both repressed and induced Hog1$^{(2)}$-dependent modules). The Hog1 functional modes can be explained in many ways, such as distinct Hog1 activity as a TF (in the nucleus) and as a kinase (in the cytoplasm), or differences in activity of other mediators, e.g., nuclear translocators or phosphatases.

Transcriptional feedback in the osmotic response network

Many components of the osmotic and mating MAPK pathways were included in modules, thereby forming both feedback and feedforward loops (Fig. 7). The algorithm predicts that the expression of *SHO1* is down-regulated by the MAPK Hog1, suggesting down-regulation of one arm of the HOG pathway upon osmotic shock. In the nucleus, active Hog1 interacts with the Msn1 transcription activator, the Rpd3 histone deacetylase, and the Tup1 transcriptional cofactor, all important for activation of the response to osmotic shock (Proft and Struhl 2002; De Nadal et al. 2004). The feedforward loop predicted between Hog1 and each of these factors (exemplified in Fig. 7B on *MSN1*) may encourage transient activation signals, allowing rapid system shutdown (Shen-Orr et al. 2002).

From the refinement results described above, we concluded that Hog1 somehow prevents cross talk with the mating/pseudohyphae pathway. Consistent with this observation, the *STE7*, *STE12*, and *SHO1* genes, which are translated into components of that pathway, are down-regulated by Hog1. On the other hand, the phosphatase Ptp3 inactivates the mating kinase Fus3, and its gene *PTP3* is up-regulated by Hog1. These predictions suggest that transcription regulation is part of the mechanism by which Hog1 prevents cross talk between the MAP kinase pathways.

Ste12 up-regulates the *FUS3* and *KSS1* genes, forming a positive feedback loop (exemplified in Fig. 7B on *FUS3*) that can increase stability and reduce response time to environmental stimuli (Shen-Orr et al. 2002). We also identified a negative feedback loop via Ste12 up-regulation of *MSG5*, indicating that the pathway has also an autoregulatory deactivation mode. Note that, upon osmotic shock, all the predicted targets that are components of the mating/pseudohyphae pathway (*SHO1*, *CDC24*, *STE7*, *KSS1*, *FUS3*, *MSG5*, *PTP3*, *STE12*, and *TEC1*) behave similarly: They are expressed only in the absence of active Hog1. Yet, the expansion procedure identifies *SHO1*, *STE7*, *PTP3*, and *STE12* as Hog1-dependent, while *CDC24*, *KSS1*, *FUS3*, *MSG5*, and *TEC1* are identified as Ste12-dependent. Indeed, experimental results not used in the computational process support these predictions: Only the predicted Ste12-dependent genes are up-regulated by pheromone that specifically activates the mating pathway (Supplemental Fig. S8). Several mechanisms for the adaptive regulation of the osmolarity pathway have been described (Hohmann 2002). The results here provide additional insight on the way transcriptional regulation might take part in the osmotic adaptation.

Discussion

Signaling and transcriptional networks are intertwined and influence each other in a complex manner. In this study, focusing on the osmotic response system in *S. cerevisiae*, we show that, by

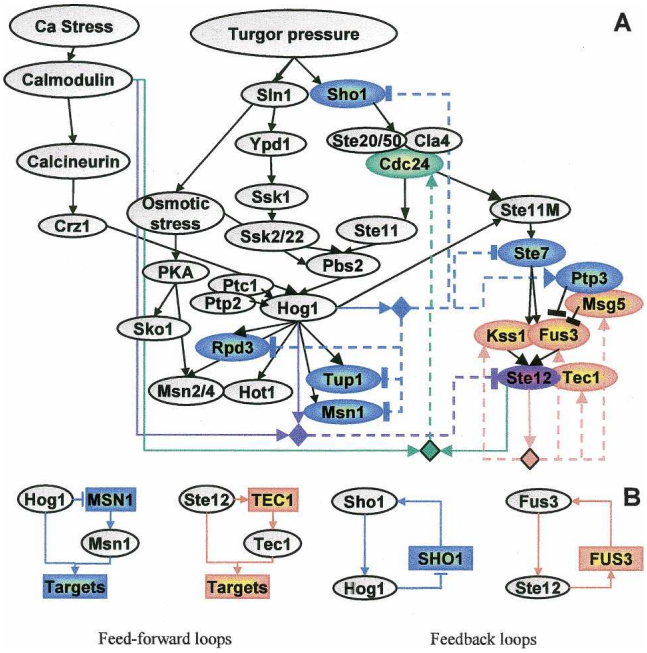


Figure 7. Complex transcriptional feedback in the yeast osmotic network model. (*A*) We highlight in color model variables whose corresponding genes were included in a module. Regulatory units are shown as diamonds, where the incoming arcs indicate the regulators they contain, and outcoming dashed arcs indicate their (direct or indirect) targets. For each regulatory unit, we use a different color for its target genes and the relevant edges. For example, the unit of Ste12 (orange) has *TEC1*, *FUS3*, *KSS1*, and *MSG5* among its targets. Unlike previous maps, the same diamond might represent several different regulatory logics, and the arcs distinguish between positive (→) and negative (⊣) feedback. The rich circuitry observed is probably part of the cellular adaptation and provides rapid and transient response to osmotic stress. (*B*) A few network motifs discovered in *A*. Rectangles indicate target genes, and ovals are proteins.

modeling together the available knowledge on signaling cascades and transcriptional regulation, we could improve our understanding of both systems in two important ways: The signaling pathways are refined based on known transcriptional regulation effects, and transcriptional regulatory modules are generated using known cascades of events along signaling pathways.

A large amount of curated qualitative knowledge on biological systems is available today. The formulation of such knowledge is shown here to be surprisingly instrumental in improving our biological understanding. Our computational framework enables modeling of the existing knowledge in the presence of feedback loops in the network, formalization of the uncertainty in this knowledge, and integration of high throughput data. In addition, the model can accommodate partial noisy measurements of diverse biological entities (Gat-Viks et al. 2006). We make major modeling simplifications: The regulatory relations are discrete logical functions, and the model describes the steady state of the system. As expected, the prediction and improvement processes that we propose here also have limitations: They are sensitive to the size and complexity of the model (e.g., number of variables, interactions, and feedback loops), the certainty in the logics, and the amount of data available. The robustness of our methods to these parameters still needs further exploration. We do have strong positive indication for the robustness of the prediction process and logical refinement procedure on small networks (Gat-Viks et al. 2006; www.cs.tau.ac.il/~rshamir/metareg). The robustness of the expansion procedure is yet needed to be systematically explored, although the biological validations in this study are highly promising. In the future, we hope this study will lead to creation of more sophisticated mathematical models and robust improvement algorithms for the analysis of genome-wide datasets.

A key advantage of our module identification approach is that we use a discriminative scoring scheme which specifically identifies modules along with their model regulators. Consequently, we can detect modules on a finer level than was previously possible (for example, novel HOG pathway-dependent repressed modules). Our method outperforms extant methods mainly because it exploits prior knowledge on the signaling pathways and on the experimental procedure. This prior knowledge helps to detect minute expression differences that are the result of distinct regulatory mechanisms, and thus the method can discard better differences that are due to noise. The main limitation in our module identification approach is that it requires high quality of prior knowledge on the signaling pathways, whereas many biological systems are only partially known. To overcome this obstacle, the model should be corrected by applying a refinement procedure before elucidating the modules. In the current study, we did not allow refinement steps that cause global effects, such as novel feedbacks or disconnected networks. We hope that, within the formalism of our model, it will be possible to develop techniques to handle those cases as well.

Although there is much to be developed both in the modeling and the algorithmic parts, by extending the concepts derived here, it is clear that simultaneous analysis of qualitative knowledge with high throughput data is a useful approach. The approach is applicable to other types of perturbations, such as siRNA, to other environmental conditions, such as pharmaceutical agents, and to other molecular data, such as protein activity levels measured by microarrays. High throughput phosphorylation measurements might allow an automated construction of kinase signaling modules using known signaling pathways. As new databases of curated knowledge on signaling pathway are developed (such as BioModels [Le Novere et al. 2006], Reactome [Joshi-Tope et al. 2005], and SPIKE [www.cs.tau.ac.il/~spike]), it will be easier to obtain the prior information on many biological systems and apply the methodology to them.

Methods

Model formalization

Our model consists of variables and relations among them, formulating prior knowledge. The model variables $X_1 \ldots X_n$ express diverse biological entities (e.g., mRNAs, proteins, metabolites, and phenotypes), and arcs between variables represent biological regulations (e.g., transcription and translation regulation, post-translational modifications). Each variable X_i is regulated by a regulatory unit Pa_i, i.e., the set of variables that have arcs into X_i. Each variable in Pa_i is called a regulator of X_i. Each variable can be in one of several (typically three) discrete states, and its state in any condition is assumed to be determined by its logic, i.e., a discrete function of its regulators' states in that condition. Note that this assumption implies that the relevant conditions are in steady state. In order to model our uncertainty about the prior knowledge, the logic of a variable X_i is formulated probabilistically as our belief that the variable attains a certain state given the state of its regulatory unit. It is represented by the conditional probability $\theta^i(X_i \mid Pa_i)$. This approach allows us to model uncertainty in prior biological knowledge and to distinguish between

regulatory logics that are known at high level of certainty and those that are more speculative. In practice, biological experiments provide continuous observations and we do not know in advance how to translate them into discrete states. Hence, each logical variable X_i is associated with an observed real-valued variable Y_i, and the conditional distribution $\psi^i(X_i \mid Y_i)$ specifies the probability of the variable X_i to attain a certain state given its observed real value. In this work, we discretize the observed values using a mixture of Gaussians model.

Our probabilistic model defines a Bayesian score, which evaluates the fit of the model predictions to the data, measured as the log likelihood of the data given the model:

$$\log \Pr(X, Y|Model) = \log\left(\frac{1}{Z}\prod_i \theta^i(X_i|Pa_i)\cdot\psi^i(X_i|Y_i)\right)$$

where Z is a normalization constant. The conditional probabilities θ^i are known from our prior knowledge on the biological system, and ψ is determined by maximizing a likelihood score using an EM-procedure. The ψ^i parameters depend strongly on the particular model, and thus we reoptimize them during each step of the improvement procedures. Given the probabilistic model, we predict the levels of variables (e.g., the activity level of proteins, the expression levels of mRNAs) using a standard probabilistic inference method called Loopy Belief Propagation (Kschischang and Loeliger 2001). As described in Gat-Viks et al. (2006), the above model is represented by a Bayesian network in case of acyclic dependencies, or by factor graph (Kschischang and Loeliger 2001), in the more general case where feedback loops, that are essential in many biological systems, are present.

Expression profiles

We compiled a dataset of 106 relevant transcription profiles selected from four large-scale studies (Gasch et al. 2000; Harris et al. 2001; Yoshimoto et al. 2002; O'Rourke and Herskowitz 2004). In addition to gene expression measurements, for each profile the experimental procedure is recorded, i.e., the environmental conditions and the genetic perturbations in the experiment. This information is used for generating model predictions. The complete list of conditions and their experimental procedures are available in Supplement B. The analysis was applied on 5700 genes that were measured in at least 100 of the conditions.

Model refinements

The refinement procedure searches for a structure modification (an added arc in the network with an accompanying logic) that improves the model significantly. Each such modified model is evaluated by the fit of its predictions to the data, measured by the Bayesian score. The score is computed by an EM-algorithm that locally maximizes the free parameters of the model: the discretization parameters ψ^i and the logic of the new regulation (Gat-Viks et al. 2006). To evaluate the significance of the improvement achieved by a particular modification to the model, we compared the likelihood scores distributions (across the 106 profiles) of the original and the modified model. The null hypothesis assumes that both models provide equal scores in each condition. The alternative hypothesis suggests higher scores for the modified model. The improvement score is the *P*-value generated using non-parametric paired Wilcoxon test. All *P*-values presented are Bonferroni corrected. The same improvement score was used for learning the regulatory logics of the six known modules.

Identification of transcriptional modules

We consider all possible regulatory units of one or two regulators out of twelve candidate regulators. These regulators include two environmental stimuli variables (Calcium stress and Turgor pressure) and 10 signaling network variables (Supplement Fig. S2). Note that the regulatory units are of two types: Variables governed by units that consist only of environmental stimuli are not affected by genetic perturbations in the model, and thus will be called model-independent modules (and their genes will be called model-independent genes). In contrast, the model-dependent modules (which contain model-dependent genes) are controlled by at least one signaling network regulator and thus influenced by genetic perturbations of model components (Fig. 2B).

Our expansion procedure seeks for each candidate gene the unit that governs it based on an improvement score. In particular, given a target gene and its candidate regulatory unit, the procedure applies a greedy search in the space of regulatory logics and discretization parameters using an EM-like procedure in order to achieve a locally maximum Bayesian score. When assigning genes to regulatory units, one should take caution about model dependence decision. Many of the reactions observed in stress and perturbation conditions can be attributed to general stress response, even if they match model predictions (Supplemental Fig. S4). To specifically discriminate model-dependent genes from model-independent genes, we require that they should be predicted significantly better by some model-dependent module than using model-independent ones. Mathematically, we define the improvement score obtained by a gene assignment to a regulatory unit as the difference between its original Bayesian score and the best model-independent Bayesian score obtained for the same gene. This approach can be viewed as hypothesis testing, where the null hypothesis is a model-independent response, and we reject it only if the alternative model-dependent hypothesis is much more convincing.

In practice, 71% of the genome (4051 genes) attained significant Bayesian score in either a model-dependent fashion (68.2%, 3887 genes) or a model-independent one (51.5%, 2935 genes) (we used a cutoff of 0.1 computed based on the shuffled data, see Supplemental Fig. S4); 876 genes (15.3%) that obtained improvement score $\geq$10 were used to construct model-dependent modules.

Our analysis is focused on model-dependent modules, but the expansion algorithm outputs also model-independent modules. Supplemental Figure S9 exemplifies one such module, which is strongly repressed by hyper-osmotic stress and enriched with ribosomal proteins. Indeed, the expression of the module genes appears by and large unaffected by the genetic perturbations in our dataset.

Module significance

To evaluate modules' significance, we tested for enrichment (hyper-geometric *P*-value) of each module's genes in each of the sets of TF targets (identified at $P < 0.01$) reported in Harbison et al. (2004) (Supplemental Fig. S6A). In addition, enrichment was computed on up-regulated and down-regulated gene sets in independent expression profiles from Gasch et al. (2000) (excluding the conditions included in the training data, and all other hyper-osmotic conditions and genetic perturbations in model variables, Supplemental Fig. S6B). Separation between modules was computed by KS-test for the difference in the expression profile distributions of each module across the same independent conditions (Supplemental Fig. S6C). All *P*-values presented are Bonferroni corrected.

Promoter analysis

We performed promoter analysis on the set of target genes in each module, aiming to find regulatory signals and putative transcription factor binding sites. For each set we searched the 500 bp

upstream of the transcription start site in each gene using Amadeus motif finder (Halperin et al. 2006). Amadeus performs de novo search for enriched motifs and also compares the motifs found to the known ones in the TRANSFAC version 8.3 database (Matys et al. 2003). The discovered motifs are listed in Supplement D.

Acknowledgments

R.S. was supported in part by the EMI-CD project that is funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health," contract number LSHG-CT-2003-503269.

References

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., et al. 2003. Computational discovery of gene modules and regulatory networks. *Nat. Biotechnol.* **21:** 1337–1342.

Beer, M.A. and Tavazoie, S. 2004. Predicting gene expression from sequence. *Cell* **117:** 185–198.

Calvano, S.E., Xiao, W., Richards, D.R., Felciano, R.M., Baker, H.V., Cho, R.J., Chen, R.O., Brownstein, B.H., Cobb, J.P., Tschoeke, S.K., et al. 2005. A network-based analysis of systemic inflammation in humans. *Nature* **437:** 1032–1037.

Covert, M.W., Knight, E.M., Reed, J.L., Herrgard, M.J., and Palsson, B.O. 2004. Integrating high-throughput and computational data elucidates bacterial networks. *Nature* **429:** 92–96.

De Nadal, E., Zapater, M., Alepuz, P.M., Sumoy, L., Mas, G., and Posas, F. 2004. The MAPK Hog1 recruits Rpd3 histone deacetylase to activate osmoresponsive genes. *Nature* **427:** 370–374.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* **95:** 14863–14868.

Friedman, N. 2004. Inferring cellular networks using probabilistic graphical models. *Science* **303:** 799–805.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7:** 601–620.

Gardner, T.S., Bernardo, D., Collins, J.J., and Lorenz, D. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* **301:** 102–105.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. 2000. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell* **11:** 4241–4257.

Gat-Viks, I., Tanay, A., and Shamir, R. 2004. Modeling and analysis of heterogeneous regulation in biological networks. *J. Comput. Biol.* **11:** 1034–1049.

Gat-Viks, I., Tanay, A., Raijman, D., and Shamir, R. 2006. A probabilistic methodology for integrating knowledge and experiments on biological networks. *J. Comput. Biol.* **13:** 165–181.

Halperin, Y., Linhart, C., Weber, G., and Shamir, R. 2006. The Amadeus motif discovery tool. RECOMB poster session. www.cs.tau.ac.il/~rshamir/amadeus/.

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., et al. 2004. Transcriptional regulatory code of a eukaryotic genome. *Nature* **431:** 99–104.

Harris, K., Lamson, R.E., Nelson, B., Hughes, T.R., Marton, M.J., Roberts, C.J., Boone, C., and Pryciak, P.M. 2001. Role of scaffolds in MAP kinase pathway specificity revealed by custom design of pathway-dedicated signaling proteins. *Curr. Biol.* **11:** 1815–1824.

Herrgard, M.J., Lee, B.S., Portnoy, V., and Palsson, B.O. 2006. Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in *Saccharomyces cerevisiae*. *Genome Res.* **16:** 627–635.

Hohmann, S. 2002. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol. Mol. Biol. Rev.* **66:** 300–372.

Hughes, T.R., Marton, M.J., Jones, A.R., Roberts, C.J., Stoughton, R., Armour, C.D., Bennett, H.A., Coffey, E., Dai, H., He, Y.D., et al. 2000. Functional discovery via a compendium of expression profiles. *Cell* **102:** 109–126.

Ideker, T., Thorsson, V., Ranish, J.A., Christmas, R., Buhler, J., Eng, J.K., Bumgarner, R., Goodlett, D.R., Aebersold, R., and Hood, L. 2001. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science* **292:** 929–934.

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., and Barkai, N. 2002. Revealing modular organization in the yeast transcriptional network. *Nat. Genet.* **31:** 370–377.

Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G.R., Wu, G.R., Matthews, L., et al. 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Res.* **33 (Database issue):** D428–D432.

Klipp, E., Nordlander, B., Krüger, R., Gennemark, P., and Hohmann, S. 2005. Integrative model of the response of yeast to osmotic shock. *Nat. Biotechnol.* **23:** 975–982.

Kschischang, F.R. and Loeliger, H.A. 2001. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47:** 498–519.

Le Novere, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., et al. 2006. BioModels Database: A free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.* **34 (Database issue):** D689–D691.

Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A.E., and Kel-Margoulis, O.V. 2003. TRANSFAC: Transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* **31:** 374–378.

Nevitt, T., Pereira, J., Azevedo, D., Guerreiro, P., and Rodrigues-Pousada, C. 2004. Expression of YAP4 in *Saccharomyces cerevisiae* under osmotic stress. *Biochem. J.* **379:** 367–374.

Ohya, Y., Kawasaki, H., Suzuki, K., Lodesborough, J., and Anraku, Y. 1991. Two yeast genes encoding calmodulin-dependent protein kinases: Isolation, sequencing and bacterial expression of CMK1 and CMK2. *J. Biol. Chem.* **266:** 12784–12794.

O'Rourke, S.M. and Herskowitz, I. 1998. The Hog1 MAP kinase prevents cross talk between the HOG and pheromone response MAP kinase pathways in *Saccharomyces cerevisiae*. *Genes & Dev.* **12:** 2874–2886.

O'Rourke, S.M. and Herskowitz, I. 2004. Unique and redundant roles for HOG MAPK pathway components as revealed by whole-genome expression analysis. *Mol. Biol. Cell* **15:** 532–542.

Proft, M. and Struhl, K. 2002. Hog1 kinase converts the Sko1-Cyc8-Tup1 repressor complex into an activator that recruits SAGA and SWI/SNF in response to osmotic stress. *Mol. Cell* **9:** 1307–1317.

Rep, M., Krantz, M., Thevelein, J.M., and Hohmann, S. 2000. The transcriptional response of *Saccharomyces cerevisiae* to osmotic shock. Hot1p and Msn2p/Msn4p are required for the induction of subsets of high osmolarity glycerol pathway-dependent genes. *J. Biol. Chem.* **275:** 8290–8300.

Sachs, K., Gifford, D., Jaakkola, T., Sorger, P., and Lauffenburger, D.A. 2002. Bayesian network approach to cell signaling pathway modeling. *Sci. STKE* **2002:** PE38.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. 2003. Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.* **34:** 166–176.

Shen-Orr, S.S., Milo, R., Mangan, S., and Alon, U. 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* **31:** 64–68.

Shitamukai, A., Hirata, D., Sonobe, S., and Miyakawa, T. 2004. Evidence for antagonistic regulation of cell growth by the calcineurin and high osmolarity glycerol pathways in *Saccaromices cerevisiaie*. *J. Biol. Chem.* **279:** 3651–3661.

Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., and Miyano, S. 2003. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics* **S2:** II227–II236.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., and Church, G.M. 1999. Systematic determination of genetic network architecture. *Nat. Genet.* **22:** 281–285.

Van Wuytswinkel, O., Reiser, V., Siderius, M., Kelders, M.C., Ammerer, G., Ruis, H., and Mager, W.H. 2000. Response of *Saccharomyces cerevisiae* to severe osmotic stress: Evidence for a novel activation mechanism of the HOG MAP kinase pathway. *Mol. Microbiol.* **37:** 382–397.

Yeang, C.H., Mak, H.C., McCuine, S., Workman, C., Jaakkola, T., and Ideker, T. 2005. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biol.* **6:** R62.

Yoshimoto, H., Saltsman, K., Gasch, A.P., Li, H.X., Ogawa, N., Botstein, D., Brown, P.O., and Cyert, M.S. 2002. Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in *Saccharomyces cerevisiae*. *J. Biol. Chem.* **277:** 31079–31088.

Yuzyuk, T., Foehr, M., and Amberg, D.C. 2002. The MEK kinase Ssk2p promotes actin cytoskeleton recovery after osmotic stress. *Mol. Biol. Cell* **13:** 2869–2880.

Received July 11, 2006; accepted in revised form November 29, 2006.

Chapter 3

Discussion

In this thesis we describe our study on molecular networks: their mathematical modeling, and the algorithmic aspects of learning them from data. We specifically focused on mathematical models for transcriptional programs, signaling cascades and metabolic pathways. We built understanding of specific cellular systems using results from high throughput measurements (mainly gene expression profiles from microarrays). The research in this thesis integrates concepts from biology, computer science, and statistics. We approached the problems from the computer science perspective, and then analyzed real biological data to demonstrate the biological implications of our methods. Moreover, we demonstrated the advantages of our methodology over extant methods. In the future, we hope that this study will lead to creation of more sophisticated and practical analysis tools of genome-wide datasets.

Computational models should provide a comprehensive description of the cellular response to intra-cellular and extra-cellular changes. In this thesis, we developed mathematical models for biological molecular networks and provided algorithms for model reconstruction from large scale experimental data. We aimed to develop a predictive model - one that can predict the effect of genetic perturbation and environmental cues correctly. In most of our works, we applied four steps of model reconstruction, each of which raises critical questions:

(i) Define the class of models and obtain a solution space. (How does one choose the most appropriate level of detail of the model?)

(ii) Score a candidate model. (How well does the model fit the data?)

(iii) Reconstruct the network by searching the solution space for the model with the best score.

(iv) Test the statistical validity of the results.

In Sections 3.1-3.4 below we describe the contributions of this thesis to each of these four steps. Practically, molecular networks are constructed in several iterations of laboratory experimentation and computational analyses. In Section 3.5 we describe our iterative approach to reconstruct and test a molecular network.

3.1. Scope and level of detail of the mathematical model

There are many challenges in modeling biological systems. How do we abstract the problem? What level of detail is necessary to understand a given phenomenon? Which prior knowledge should be used and how to incorporate it into the model? There is no unique answer to any of these questions.

Many mathematical predictive models have been proposed at various levels of granularity, ranging from discrete models to differential equations models. In all levels of resolution, a key obstacle in trying to reconstruct a predictive model from data is the large solution space. Thus, the solution space is often limited using prior biological knowledge (see Section 1.3.3 for details). In our research, we developed two modeling approaches: chain functions (Chapters 2.2 and 2.3), and the Metareg methodology (Chapters 2.4, 2.5, 2.6). Both models integrate various types of qualitative prior knowledge about mechanism of regulation. The chain functions model is dramatically constrained by involving a general logic and structure observed in many real biological regulation functions. In contrast, the MetaReg model utilizes knowledge about the logic and topology of specific reaction mechanisms.

The **chain function model** (Chapter 2.2) is a deterministic Boolean model. In this model, the state of the target gene depends on the influence of its direct regulator, whose activity may in turn depend on the influence of another regulator, and so on in a chain of dependencies. This model assumes that each target genes is learned independently of other genes. In a subsequent study (Chapter 2.3), we further improved the model to reflect regulation functions that combine several chains. We showed that these functions reflect common biological regulation behavior, and often occur in networks. We proved that the number of chain functions with n control variables is exponentially smaller than the total number of Boolean functions. Hence, the size of the search space is reduced exponentially. We applied our approach to transcription profiles of the yeast galactose pathway and demonstrated the improved accuracy obtained by using chain functions instead of searching through all Boolean functions.

Next, we proposed the Metareg model for formalizing of prior qualitative knowledge on biological networks. Most regulatory models in the literature include only one type of regulatory components (e.g., genes in [52], proteins in [65, 66]). Instead, our model can contain heterogeneous biological components (such as mRNA, proteins, and metabolites). Each of the components is associated with a discrete regulation function. Consequently, our model can express the environmental conditions and can capture

diverse logical relations on several regulatory levels (metabolism, transcription, translation, post-translation, and feedback loops among them). We developed a deterministic model formulation (Chapter 2.4) and generalized it to a probabilistic formulation (Chapters 2.5 and 2.6). Our probabilistic approach allows us to model uncertainty in prior biological knowledge, and to distinguish between regulatory relations that are known at high level of certainty and those that are only hypothesized. The probabilistic model also allows us to mix noisy continuous measurements with discrete regulatory logic. Unlike the commonly used Bayesian network model, our model (which is a factor graph [74]) can directly accommodate steady state (undelayed) feedback loops.

The MetaReg model is predictive and as such it computes the expected level of each component in each condition. In the deterministic formulation, we proved that computing model predictions is hard in the biologically relevant case where the network contains cycles. Hence, we provided a practical methodology for prediction based on approximations for the minimum feedback set problem (Chapter 2.4). In the probabilistic setting, the predictions were computed by inference algorithms. Inference in graphical models is an NP-hard problem that was extensively studied. We developed an instantiation-based inference algorithm that exploits the special characteristics of the biological network and achieves a dramatic reduction of the time complexity. Using simulations, we studied the performance of our inference algorithm and of several other algorithms on the Metareg model and showed that we obtain a reliable inference even in the presence of feedback loops and complex logic (Chapter 2.5).

3.2. Evaluation of a candidate network in accordance to data

In order to perform network reconstruction, or any other kind of comparison of putative solutions, one has to examine numerous candidate networks and determine the most appropriate one. This requires a way to score a specific network vis-à-vis the available data. The standard scoring technique is to compare model predictions with measurements, and assess how well do the network fits the data. If model predictions are consistent with data, the network is adequately characterized. If there are discepancies, the model should be refined to fit the data. Most scoring approaches evaluate the *consistency* score, for example the percentage of agreement [76], the p-value of maximum agreement [78], or the mutual information between predictions and observations. In probabilistic models, the likelihood of the model given the data serves as a direct measure of consistency, without the need to produce model predictions [64].

In our deterministic discrete frameworks (Chapters 2.2 and 2.4), we used a discrepancy score, a measurement of inconsistency instead of the standard consistency measures. We defined *discrepancy* as the difference (or squared difference) between model predictions and data measurements. Initially we developed the **Funcfit** score**,** which calculates the discrepancy p-value. We used the chain functions model to show the advantages of this score over extant scores (Chapter 2.2). Next, the discrepancy score was applied also for the evaluation of cyclic deterministic models (Chapter 2.4).

For our probabilistic graphical model, we used a likelihood-based fitness score (Chapters 2.5). We started with an initial model and tried to improve it by searching for the modified model with the best likelihood score. The likelihood was computed by a probabilistic inference algorithm (see Section 3.1). In a subsequent work (Chapter 2.6), we argued that high likelihood score does not necessarily represent robust reconstruction. Instead, we proposed a discriminative likelihood ratio score called **improvement score**, which compares the likelihood of a candidate model vs. a null hypothesis model. By using this score, we were able to identify significant and robust model improvements.

In order to evaluate only the structure of the model, without attempting to evaluate the regulation functions, the scoring scheme is substantially different from the approach described above. The common measure is the mutual information between the observations of the parents and observations of the target (e.g., [60, 64, 79]). However, the mutual information is very sensitive to over-fitting. To address this problem, we developed the **regSpec** score (Chapter 2.2), which is essentially an approximate p-value of the mutual information.

3.3. Reconstruction of the network

Learning the network model is often cast as an optimization problem, where the computational task is to search in the solution space and find a solution of maximum score. This optimization problem is often addressed using standard heuristic search techniques (e.g., [64, 71]). Due to lack of comprehensive data, most of the practical applications focus on particular sub-networks, particular modules or pathways (e.g., [65, 77, 80, 81]).

A relatively simple reconstruction goal is to optimize only single regulation function. We showed that this function optimization problem is NP-hard in the discrete framework of Metareg. Hence, we translated the problem to a combinatorial problem on matrices, and provided a polynomial-time, constant factor approximation for learning the

regulation of a single entity. The strategy was tested on the lysine biosynthesis pathway in yeast (Chapter 2.4). A similar approximation was used for learning regulation functions in the probabilistic framework (Chapters 2.5).

The more complicated reconstruction goal is to compute both structure and logic together. The thesis research was started by analyzing the yeast galactose pathway, assuming that its regulatory mechanism belongs to the chain functions model (Chapter 2.2). Due to the reduced size of the search space, it was possible to perform an exhaustive search through all the solution space to find the solution with the best score. Next, in the probabilistic Metareg model, we searched exhaustively for single-edge structural modifications, but for each possible candidate edge, we had to re-optimize the regulation functions and the missing parameters using an EM-procedure. This approach was tested on the osmotic response network in yeast (Chapter 2.5 and 2.6).

3.4. Statistical significance of the results

A major challenge is to infer robust computational models and to be able to evaluate the significance of the conclusions. Models with overabundance of potential structures and parameters are at the risk of over-fitting and of non-specific predictions. Moreover, learning algorithms might generate high rate of false positive results due to multiple comparisons. Clustering and network reconstruction algorithms are commonly used in bioinformatics analysis, but there are no agreed upon guidelines for statistical evaluation of their results. In this thesis, we developed statistical methods for the assessment of clustering models (Chapter 2.1) and evaluation of network features (Chapters 2.5 and 2.6).

In Chapter 2.1, we devised a statistically-based method for the evaluation of a clustering model according to prior qualitative biological knowledge. Given a vector of (continuous or discrete) functional attributes for each gene (e.g., taken from the Gene Ontology database [82]), our method tests the dependency between the attributes and the grouping of the genes. The test can be applied simultaneously to all the attributes. We validated our approach using simulated data and showed that our scoring method outperforms several extant methods.

In Chapters 2.5 and 2.6, we assigned statistical meaning to learned features in our network model. We derived p-values for the features using the standard bootstrap method. In addition, we developed method for sampling from the network model, and used it to compute p-values using a direct likelihood ratio test. We evaluated the accuracy

of our approach using ROC analysis on simulated data and via cross validation tests on empirical data. Finally, the biological conclusions were supported by several independent experimental data sets.

3.5. An iterative reconstruction of molecular networks.

In real life science settings, reconstruction of molecular networks is an iterative process that involves both computational analysis and laboratory experimentation [36, 83-85]. In chapter 2.3, we developed an experimental design approach for the reconstruction of molecular networks in an iterative manner. Our algorithms perform de-novo reconstruction of the model using a minimal number of experiments and genetic perturbations, assuming accurate experimental results and the chain functions model. We developed optimal iterative reconstruction schemes for several scenarios.

In chapters 2.4, 2.5, 2.6 our approach was different: We started with an initial mathematical model of the system based on well-established available knowledge. The model produces predictions (usually by simulations) on the behavior of the system, which are compared with the experimental measurements. The mismatches between predictions and observations were used in order to correct the model computationally. The computational hypotheses should be validated in the laboratory, and the process can iterate until an adequate model is obtained. We programmed a visual software application that performs all these computational steps from model construction to generation of hypotheses [86]. This work is not included in this thesis.

Bibliography

1. Gat-Viks, I., R. Sharan, and R. Shamir, *Scoring clustering solutions by their biological relevance.* Bioinformatics, 2003. **19**(18): p. 2381-9.
2. Gat-Viks, I. and R. Shamir, *Chain functions and scoring functions in genetic networks.* Bioinformatics, 2003. **19 Suppl 1**: p. i108-17.
3. Gat-Viks, I., et al., *Reconstructing chain functions in genetic networks.* Pac Symp Biocomput, 2004: p. 498-509.
4. Gat-Viks, I., et al., *Reconstructing chain functions in genetic networks.* SIAM Journal of Discrete Mathematics, 2006. **20**: p. 727-740.
5. Gat-Viks, I., A. Tanay, and R. Shamir, *Modeling and analysis of heterogeneous regulation in biological networks.* Proceedings of the first RECOMB satellite workshop on Regulatory Genomics, E. Eskin and C. Workman (editors), Lecture Notes in Bioinformatics, Vol. 3318 pp. 98--113, Springer, Berlin. , 2005.
6. Gat-Viks, I., A. Tanay, and R. Shamir, *Modeling and analysis of heterogeneous regulation in biological networks.* J Comput Biol, 2004. **11**(6): p. 1034-49.
7. Gat-Viks, I., et al., *A probabilistic methodology for integrating knowledge and experiments on biological networks.* Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 05), pp. 31-47, Lecture Notes in Bioinformatics 3500, Springer, Berlin, 2005. , 2005.
8. Gat-Viks, I., et al., *A probabilistic methodology for integrating knowledge and experiments on biological networks.* J Comput Biol, 2006. **13**(2): p. 165-81.
9. Gat-Viks, I. and R. Shamir, *Refinement and expansion of signaling pathways: the osmotic response network in yeast.* Genome Research, 2007(to appear).
10. Jeong, H., et al., *The large-scale organization of metabolic networks.* Nature, 2000. **407**(6804): p. 651-4.
11. Milo, R., et al., *Superfamilies of evolved and designed networks.* Science, 2004. **303**(5663): p. 1538-42.
12. Yeger-Lotem, E., et al., *Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction.* Proc Natl Acad Sci U S A, 2004. **101**(16): p. 5934-9.
13. Sharan, R., I. Ulitsky, and R. Shamir, *Functional analysis of protein interaction networks.* 2007: p. To appear in Molecular Systems Biology.
14. Tavazoie, S., et al., *Systematic determination of genetic network architecture.* Nat Genet, 1999. **22**(3): p. 281-5.
15. Aldridge, B.B., et al., *Physicochemical modelling of cell signalling pathways.* Nat Cell Biol, 2006. **8**(11): p. 1195-203.
16. Friedman, N., *Inferring cellular networks using probabilistic graphical models.* Science, 2004. **303**(5659): p. 799-805.
17. Janes, K.A. and M.B. Yaffe, *Data-driven modelling of signal-transduction networks.* Nat Rev Mol Cell Biol, 2006. **7**(11): p. 820-8.

18. Gasch, A.P., et al., *Genomic expression programs in the response of yeast cells to environmental changes.* Mol Biol Cell, 2000. **11**(12): p. 4241-57.
19. Hughes, T.R., et al., *Functional discovery via a compendium of expression profiles.* Cell, 2000. **102**(1): p. 109-26.
20. Golub, T.R., et al., *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.* Science, 1999. **286**(5439): p. 531-7.
21. Beer, M.A. and S. Tavazoie, *Predicting gene expression from sequence.* Cell, 2004. **117**(2): p. 185-98.
22. Eisen, M.B., et al., *Cluster analysis and display of genome-wide expression patterns.* Proc Natl Acad Sci U S A, 1998. **95**(25): p. 14863-8.
23. Patterson, S.D. and R.H. Aebersold, *Proteomics: the first decade and beyond.* Nat Genet, 2003. **33 Suppl**: p. 311-23.
24. Dunn, W.B., N.J. Bailey, and H.E. Johnson, *Measuring the metabolome: current analytical technologies.* Analyst, 2005. **130**(5): p. 606-25.
25. Buck, M.J. and J.D. Lieb, *ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments.* Genomics, 2004. **83**(3): p. 349-60.
26. Mockler, T.C., et al., *Applications of DNA tiling arrays for whole-genome analysis.* Genomics, 2005. **85**(1): p. 1-15.
27. Fields, S., *High-throughput two-hybrid analysis. The promise and the peril.* Febs J, 2005. **272**(21): p. 5391-9.
28. Tong, A.H., et al., *Systematic genetic analysis with ordered arrays of yeast deletion mutants.* Science, 2001. **294**(5550): p. 2364-8.
29. Giaever, G., et al., *Functional profiling of the Saccharomyces cerevisiae genome.* Nature, 2002. **418**(6896): p. 387-91.
30. Caenepeel, S., et al., *The mouse kinome: discovery and comparative genomics of all mouse protein kinases.* Proc Natl Acad Sci U S A, 2004. **101**(32): p. 11707-12.
31. Manning, G., et al., *The protein kinase complement of the human genome.* Science, 2002. **298**(5600): p. 1912-34.
32. Tamayo, P., et al., *Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.* Proc Natl Acad Sci U S A, 1999. **96**(6): p. 2907-12.
33. Ben-Dor, A., R. Shamir, and Z. Yakhini, *Clustering gene expression patterns.* J Comput Biol, 1999. **6**(3-4): p. 281-97.
34. Sharan, R. and R. Shamir, *CLICK: a clustering algorithm with applications to gene expression analysis.* Proc Int Conf Intell Syst Mol Biol, 2000. **8**: p. 307-16.
35. Sabatti, C., et al., *Co-expression pattern from DNA microarray experiments as a tool for operon prediction.* Nucleic Acids Res, 2002. **30**(13): p. 2886-93.
36. Ideker, T., et al., *Integrated genomic and proteomic analyses of a systematically perturbed metabolic network.* Science, 2001. **292**(5518): p. 929-34.
37. Spellman, P.T., et al., *Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.* Mol Biol Cell, 1998. **9**(12): p. 3273-97.

38. Cheng, Y. and G.M. Church, *Biclustering of expression data.* Proc Int Conf Intell Syst Mol Biol, 2000. **8**: p. 93-103.
39. Tanay, A., R. Sharan, and R. Shamir, *Discovering statistically significant biclusters in gene expression data.* Bioinformatics, 2002. **18 Suppl 1**: p. S136-44.
40. Ihmels, J., et al., *Revealing modular organization in the yeast transcriptional network.* Nat Genet, 2002. **31**(4): p. 370-7.
41. Tompa, M., et al., *Assessing computational tools for the discovery of transcription factor binding sites.* Nat Biotechnol, 2005. **23**(1): p. 137-44.
42. Segal, E., et al., *A module map showing conditional activity of expression modules in cancer.* Nat Genet, 2004. **36**(10): p. 1090-8.
43. Tanay, A., et al., *Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data.* Proc Natl Acad Sci U S A, 2004. **101**(9): p. 2981-6.
44. Huang, E., et al., *Gene expression phenotypic models that predict the activity of oncogenic pathways.* Nat Genet, 2003. **34**(2): p. 226-30.
45. Mootha, V.K., et al., *PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes.* Nat Genet, 2003. **34**(3): p. 267-73.
46. Bailey, T.L. and C. Elkan, *Fitting a mixture model by expectation maximization to discover motifs in biopolymers.* Proc Int Conf Intell Syst Mol Biol, 1994. **2**: p. 28-36.
47. Kellis, M., et al., *Sequencing and comparison of yeast species to identify genes and regulatory elements.* Nature, 2003. **423**(6937): p. 241-54.
48. Tanay, A., I. Gat-Viks, and R. Shamir, *A global view of the selection forces in the evolution of yeast cis-regulation.* Genome Res, 2004. **14**(5): p. 829-34.
49. Xie, X., et al., *Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals.* Nature, 2005. **434**(7031): p. 338-45.
50. Bussemaker, H.J., H. Li, and E.D. Siggia, *Regulatory element detection using correlation with expression.* Nat Genet, 2001. **27**(2): p. 167-71.
51. Conlon, E.M., et al., *Integrating regulatory motif discovery and genome-wide expression analysis.* Proc Natl Acad Sci U S A, 2003. **100**(6): p. 3339-44.
52. Bar-Joseph, Z., et al., *Computational discovery of gene modules and regulatory networks.* Nat Biotechnol, 2003. **21**(11): p. 1337-42.
53. Tanay, A. and R. Shamir, *Multilevel modeling and inference of transcription regulation.* J Comput Biol, 2004. **11**(2-3): p. 357-75.
54. Sharan, R. and T. Ideker, *Modeling cellular machinery through biological network comparison.* Nat Biotechnol, 2006. **24**(4): p. 427-33.
55. Jansen, R., et al., *A Bayesian networks approach for predicting protein-protein interactions from genomic data.* Science, 2003. **302**(5644): p. 449-53.
56. Matthews, L.R., et al., *Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "interologs".* Genome Res, 2001. **11**(12): p. 2120-6.
57. Kelley, R. and T. Ideker, *Systematic interpretation of genetic interactions using protein networks.* Nat Biotechnol, 2005. **23**(5): p. 561-6.

58. Ideker, T.E., V. Thorsson, and R.M. Karp, *Discovery of regulatory interactions through perturbation: inference and experimental design.* Pac Symp Biocomput, 2000: p. 305-16.
59. Akutsu, T., S. Miyano, and S. Kuhara, *Identification of genetic networks from a small number of gene expression patterns under the Boolean network model.* Pac Symp Biocomput, 1999: p. 17-28.
60. Liang, S., S. Fuhrman, and R. Somogyi, *Reveal, a general reverse engineering algorithm for inference of genetic network architectures.* Pac Symp Biocomput, 1998: p. 18-29.
61. Thieffry, D. and R. Thomas, *Qualitative analysis of gene networks.* Pac Symp Biocomput, 1998: p. 77-88.
62. Dhaeseleer, P., et al., *Linear modeling of mRNA expression levels during CNS development and injury.* Proceedings of the 1999 Pacific Symposioum in Biocomputing (PSB 99), 1999: p. 41–52.
63. Ronen, M., et al., *Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics.* Proc Natl Acad Sci U S A, 2002. **99**(16): p. 10555-60.
64. Friedman, N., et al., *Using Bayesian networks to analyze expression data.* J Comput Biol, 2000. **7**(3-4): p. 601-20.
65. Sachs, K., et al., *Causal protein-signaling networks derived from multiparameter single-cell data.* Science, 2005. **308**(5721): p. 523-9.
66. Kim, S.Y., S. Imoto, and S. Miyano, *Inferring gene networks from time series microarray data using dynamic Bayesian networks.* Brief Bioinform, 2003. **4**(3): p. 228-35.
67. Gardner, T.S., et al., *Inferring genetic networks and identifying compound mode of action via expression profiling.* Science, 2003. **301**(5629): p. 102-5.
68. Segal, E., et al., *Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data.* Nat Genet, 2003. **34**(2): p. 166-76.
69. Imoto, S., et al., *Combining microarrays and biological knowledge for estimating gene networks via bayesian networks.* J Bioinform Comput Biol, 2004. **2**(1): p. 77-98.
70. Herrgard, M.J., et al., *Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in Saccharomyces cerevisiae.* Genome Res, 2006. **16**(5): p. 627-35.
71. Klipp, E., et al., *Integrative model of the response of yeast to osmotic shock.* Nat Biotechnol, 2005. **23**(8): p. 975-82.
72. Hoffmann, A., et al., *The IkappaB-NF-kappaB signaling module: temporal control and selective gene activation.* Science, 2002. **298**(5596): p. 1241-5.
73. Schoeberl, B., et al., *Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors.* Nat Biotechnol, 2002. **20**(4): p. 370-5.
74. Kschischang, F., B. Frey, and H. Loeliger, *Factor graphs and the sum-product algorithm* IEEE Trans. Inform. Theory, 1998.

75. Yeang, C.H., T. Ideker, and T. Jaakkola, *Physical network models.* J Comput Biol, 2004. **11**(2-3): p. 243-62.
76. Covert, M.W., et al., *Integrating high-throughput and computational data elucidates bacterial networks.* Nature, 2004. **429**(6987): p. 92-6.
77. Kumar, N., et al., *Applying computational modeling to drug discovery and development.* Drug Discov Today, 2006. **11**(17-18): p. 806-11.
78. Tanay, A. and R. Shamir, *Computational expansion of genetic networks.* Bioinformatics, 2001. **17 Suppl 1**: p. S270-8.
79. Pe'er, D., A. Regev, and A. Tanay, *Minreg: inferring an active regulator set.* Bioinformatics, 2002. **18 Suppl 1**: p. S258-67.
80. Lee, E., et al., *The roles of APC and Axin derived from experimental and theoretical analysis of the Wnt pathway.* PLoS Biol, 2003. **1**(1): p. E10.
81. Hendriks, B.S., et al., *Computational modelling of ErbB family phosphorylation dynamics in response to transforming growth factor alpha and heregulin indicates spatial compartmentation of phosphatase activity.* Syst Biol (Stevenage), 2006. **153**(1): p. 22-33.
82. Harris, M.A., et al., *The Gene Ontology (GO) database and informatics resource.* Nucleic Acids Res, 2004. **32**(Database issue): p. D258-61.
83. Workman, C.T., et al., *A systems approach to mapping DNA damage response pathways.* Science, 2006. **312**(5776): p. 1054-9.
84. Ben-Tabou de-Leon, S. and E.H. Davidson, *Deciphering the underlying mechanism of specification and differentiation: the sea urchin gene regulatory network.* Sci STKE, 2006. **2006**(361): p. pe47.
85. Mogilner, A., R. Wollman, and W.F. Marshall, *Quantitative modeling in cell biology: what is it good for?* Dev Cell, 2006. **11**(3): p. 279-87.
86. Ulitsky, I., I. Gat-Viks, and R. Shamir, *MetaReg : A tool for modeling, analysis and visualization of biological systems by using large-scale experimental data.* In preparation., 2007. www.cs.tau.ac.il/~rshamir/metareg/

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב

גישות חישוביות לשחזור רשתות ביולוגיות מולקולאריות

חיבור לשם קבלת תואר "דוקטור לפילוסופיה"

מאת **עירית גת-ויקס**

בהנחייתו של פרופ' **רון שמיר**

הוגש לסנאט של אוניברסיטת ת"א

ינואר 2007

תמצית

רשתות מולקולאריות שולטות בפעילות התאית ולכן שחזורן הוא אחד האתגרים המרכזיים בדרכה של הביולוגיה המודרנית להבין מערכות ביולוגיות מורכבות. טכנולוגיות חדישות מאפשרות מדידה של אלפי נתונים מולקולאריים בו זמנית, וניטור של הרשת המולקולארית מנקודות מבט שונות. המחקר המתואר בתזה זו עוסק ברשתות מולקולאריות, פיתוח מודלים מתמטיים לתיאור פעילותן, ובניית אלגוריתמים להסקת מודלים אלו ממדידות ביולוגיות. השיטות שפיתחנו מאפשרות הסקה מתוך מגוון רחב של מקורות מידע, הכולל עקרונות ביולוגיים כלליים, הבנה יסודית ומעמיקה של ריאקציות ביולוגיות, ומדידות ביולוגיות בקנה מידה גדול. באמצעות שימוש בשיטות אלו על מדידות ביולוגיות בשמר האופים אנו מראים שניתוח משולב של מודלים מתמטיים ונתונים ביולוגיים מאפשר חיזוי של התנהגות המערכת התאית. יכולת חיזוי זאת מאפשרת הסקת תובנות חדשות אודות המערכת הביולוגית ובניית השערות שיהוו בסיס למחקר עתידי.

תקציר

רקע כללי

שיטות ביו-טכנולוגיות מאפשרות מדידה של אלפי מולקולות ביולוגיות בו-זמניות. נתונים אלו משקפים את ההתנהגות התאית תחת תנאים שונים. היישומים של טכנולוגיה זו הם רבים ומגוונים, וכוללים בתוכם הבנת התפקוד של מולקולות ביולוגיות, בניית רשתות של יחסים בין מולקולות ביולוגיות, דיאגנוזה של מחלות, ואפיון ההשפעות של טיפולים רפואיים. בניגוד לשיטות מסורתיות שהתמקדו במחקר גן או חלבון בודד, הטכנולוגיות החדשות מאפשרות מחקר של רכיבים רבים ושונים בתא בו זמנית, והבנה של עקרונות ותהליכים גלובליים. בעזרת טכנולוגיות אלו, בפעם הראשונה בהיסטוריה, ניתן יהיה להשיג הבנה מעמיקה וכוללת של עולם החי.

לאור התפתחות השיטות הביו-טכנולוגיות בשנים האחרונות, מופנים כעת מאמצים רבים לפיתוח מתודולוגיות חישוביות שמטרתן ניתוח והפקת ידע מהנתונים החדשים. מידע רב ממקורות שונים משולב יחדיו על מנת להשיג הבנה טובה יותר של הרשת המולקולארית. נכון להיום, למרות המחקר הרב המתבצע, ואפילו באורגניזמים נחקרים ביותר, רשתות ביולוגיות ספורות בלבד פוענחו במלואן.

על מנת להכיר את הרשתות הביולוגיות בצורה טובה יותר, יש לבנות מודל מתמטי של המערכת הביולוגית. המודל המתמטי עשוי להעניק תובנות מסוגים שונים: (1) **זיהוי עקרונות כלליים** - ניתוח כולל של המודל יכול להאיר תכונות כלליות כגון מוטיבים שכיחים ברשת [11,12]. (2) **אפיון תכונות תפקודיות** - ניבוי הפונקציה הביולוגית יכול להיעשות על בסיס המודל החישובי (ראה לדוגמא [13,14]). (3) **ניתוח התנהגות המערכת הביולוגית** - מודל חישובי יכול לצפות את מצב המערכת תחת תנאים שונים (ראה לדוגמא [15-17]). המחקר המתואר בתזה זו מתמקד בשאלות הנוגעות לניתוח התנהגות הרשת המולקולארית. מטרתנו לתכנן ולשחזר מודלים בעלי יכולת ניבוי מצבים עבור רשתות מולקולאריות.

מבין הטכנולוגיות החדישות, מדידות רחבות היקף של ביטוי גנים [18,19] הינן הפופולאריות ביותר. מדידות אלו נעשות באמצעות טכנולוגיה של שבבים ביולוגיים המאפשרת מדידה בו-זמנית של אלפי מולקולות רנ"א שליח. קיימות טכניקות רבות נוספות למדידות רחבות היקף [23-31] שנותנות מידע על מגוון סוגי מולקולות ופעילויות בתא. מידע רחב היקף זה הוא מורעש, מוטה, ובחלקו חסר, ולכן אינו מאפשר ניתוח פשוט וישיר.

קיימים שלושה סוגים ראשיים של מודלים חישוביים המיועדים לניצוח ביולוגי של תהליכים תאיים: (1) חלוקה לקבוצות (צבירים) של מולקולות ביולוגיות עם מכנה משותף פונקציונאלי, (2) מודלים המתארים אינטראקציות בין המרכיבים המולקולאריים, וכן (3) מודלים המייצגים את יחסי השליטה בין המרכיבים המולקולאריים. כל אחת משלוש הגישות מאפשרת ניתוח נתונים ביולוגיים בעזרת שילוב סוגי אינפורמציה רבים ושונים. האלגוריתמים והשיטות בהן משתמשים עבור בניית כל אחד מן המודלים מפורטות להלן.

חלוקת המערכת הביולוגית לצבירים פונקציונאליים

שלב מרכזי בניתוח מדידות ביולוגיות בהיקף נרחב הוא חלוקת המרכיבים הביולוגיים (כגון חלבונים, רנ"א שליח) לקבוצות עם תבנית התנהגות משותפת. חלוקה זו מושגת באמצעות אלגוריתמים המיועדים למצוא חלוקה מיטבית לצבירים. אלגוריתמים אלו מייצרים צבירים המאופיינים בהומוגניות בתוך הצבירים (האיברים בכל צביר דומים אחד לשני) ובהפרדה טובה בין הצבירים (איברים מצבירים שונים הינם בעלי התנהגות נבדלת). שיטות חלוקה לצבירים מופעלות על מטריצות גדולות של ביטוי גנטי ומייצרות קבוצות גנים אינפורמטיביות. פותחו אלגוריתמים רבים לחלוקה לצבירים, המתבססים על חלוקה היררכית לצבירים [22], חלוקה ישירה לקבוצות [32-34] , וחלוקה לדו-צבירים (קבוצות בעלות התנהגות משותפת רק בתת קבוצה של הניסויים) [38-40]. כמו כן, קיימות שיטות המשלבות מספר סוגי מידע על מנת לבנות חלוקה נאותה לצבירים [42,43]. על מנת להבין את המהות הפונקציונאלית של צבירי גנים, מפעילים שיטות לאיתור העשרה פונקציונאלית וזיהוי אתרי קישור של גורמי שעתוק [14,41].

ייצוג המערכת הביולוגית כרשת אינטראקציות

בשיטה זו המערכת הביולוגית מיוצגת באופן אבסטרקטי באמצעות גרף. הגרף מכיל קודקודים המייצגים מרכיבים מולקולאריים, וקשתות הגרף מייצגות אינטראקציות בין המרכיבים. קיימים שני מודלים נפוצים. המודל הראשון הינו **רשת אינטראקציות חלבון-חלבון,** בה המרכיבים המולקולאריים המיוצגים בקדקודים הם חלבונים. רשת זו משוחזרת בעזרת מדידות ביולוגיות רחבות היקף של אינטראקציות בין חלבונים, בשילוב עם סוגי מידע נוספים כגון ידע פונקציונאלי, מיקום בתא, והשוואה לאורגניזמים אחרים [54-56]. על בסיס הרשת, ניתן לשפר את רמת ההבנה לגבי תפקודם הביולוגי של חלבונים (ראה סקירה [13]) ולגבי עקרונות גלובליים [10-12,57].

מודל נוסף הוא **רשת אינטראקציות דנ"א-חלבון**. רשת זו מאפשרת ייצוג של גורמי השעתוק בתא והגנים הנשלטים על ידי כל אחד מהם, ולכן קרויה גם *רשת שעתוק*, או *רשת הבקרה על שעתוק*. באופן עקרוני הרשת מבוססת על מדידות קישור גורמי שעתוק לאתרי הכרה בדנ"א, אולם בשל רעש המדידות נהוג לשלב מידע ממקורות מידע נוספים, כגון רצפים גנומיים [46-49], ומידע על ביטוי גנים [14,50-53].

ייצוג הרשת המולקולארית כמודל מתמטי בעל יכולת ניבוי

בביולוגיה מולקולארית מסורתית נעשה שימוש נרחב בדיאגראמות המתארות מערכות ביולוגיות. דיאגראמות אלו משמשות לארגון האינפורמציה, הבנת תוצאות ניסויים, ותכנון ניסויים נוספים. בימינו, תודות לטכנולוגיות החדישות המייצרות מידע רחב היקף, נעשה בלתי אפשרי לפרש באופן ידני מדידות רחבות היקף לאור הידע הקיים על המערכת. לפיכך, גדל

הצורך לבטא את המערכת הביולוגית באופן חישובי, כמודל המסוגל לצפות את התנהגות המערכת. בעזרת השוואת ניבויי המודל למדידות במעבדה, ניתן לאמוד את טיב המודל. כמו כן, ניתן להפעיל אלגוריתמים לשיפור המודל באופן שיטתי ויעיל, על מנת להשיג התאמה טובה יותר בין המודל לניסויים. בניית מודלים מתמטיים בעלי יכולת ניבוי עבור רשתות ביולוגיות הקשורות במחלות האדם, הוא צעד חשוב והכרחי על מנת להבין מנגנוני מחלות, ועשוי להיות בעל תרומה ניכרת לתכנון תרופות ודיאגנוזה רפואית.

מודל מתמטי בעל כושר ניבוי הינו הפשטה של המציאות, שממנה ניתן לחשב את ההתנהגות הצפויה של כל אחד ממרכיבי המערכת בכל תנאי. המודל מיוצג בדרך כלל על ידי רשת, כאשר הקודקודים מייצגים רכיבים ביולוגיים, ו*מבנה* הקשתות מייצג את יחסי שליטה בין המרכיבים. לכל קודקוד מיוחסת *פונקצית בקרה (רגולציה)* המתארת את הלוגיקה של ההשפעה המכניסטית. רשתות אלו קרויות לעיתים *רשתות מולקולאריות* או *רשתות תאיות*. כמו כן, לעיתים הרשת מכונה בשם המעיד על הרכיבים בתוכה או התהליכים אותם היא מייצגת (לדוגמא, *רשתות* אותות או *רשתות מטבוליות*). הרשת יכולה להיות מיוצגת בעזרת מודלים מתמטיים בדידים או רציפים, דטרמיניסטיים או הסתברותיים, סטטיים או דינאמיים, וברמות פירוט מתמטי שונות. לדוגמא, פותחו מודלים לרשתות בדידות [58-60], רציפות [61,62], כאלה המבוססים על משוואות דיפרנציאליות [63], ומודלים הסתברותיים המיוצגים על ידי רשתות בייזיאניות [64-66].

הבעיה העיקרית בשחזור המודל היא גודל מרחב הפתרונות. קיימים יותר מידי פתרונות אפשריים, ונדרשת כמות לא ברת השגה של נתונים על מנת לזהות בביטחון את הרשת המולקולארית. על מנת לאפשר שיחזור של הרשת, מגבילים את מרחב החיפוש לפתרונות סבירים ביולוגית על בסיס ידע מוקדם. ניתן לסווג את הידע הביולוגי המוקדם שבו נהוג להשתמש לארבע קטגוריות, על בסיס סוג האינפורמציה: **(1) ידע איכותי על מבנה הרשת** - שימוש בהגבלות כלליות על מבנה הרשת שמקורן בהבנה כללית של רשתות ביולוגיות (לדוגמא, הגבלת מספר הקשתות הנכנסות לקדקוד, או הגבלות על לוגיקות הבקרה [67,68]). **(2) ידע כמותי על מבנה הרשת** - הטיה של מבנה הרשת המשוחזרת על ידי מדידות ביולוגיות רחבות היקף של אינטראקציות חלבון-חלבון וחלבון-דנ"א (ראה לדוגמא [69,70]). **(3) ידע איכותי על מנגנוני ריאקציות מולקולאריות** - זהו ידע על מהות התהליכים וההשפעות המולקולאריות, ללא אינפורמציה כמותית על פרמטרים קינטיים וקבועי ריאקציות. קיימים מספר מודלים המשלבים ידע מסוג זה [15,71-75]. **(4). ידע כמותי על מנגנוני ריאקציות מולקולאריות** - כיום הידע הכמותי הקיים הוא בעיקר אודות קבועי ריאקציות מטבוליות, ואינפורמציה זו משולבת בעיקר בשחזור רשתות מטבוליות [15,76].

תקציר המאמרים הכלולים בתזה

עבודה זו מבוססת על שבעה מאמרים, אשר פורסמו בכתבי עת מדעיים והוצגו בכנסים מדעיים. להלן פירוט תקציר המאמרים:

1. Scoring clustering solutions by their biological relevance.
Irit Gat-Viks, Roded Sharan and Ron Shamir.
Published in *Bioinformatics* [1].

שלב מרכזי באנליזה של מדידות ביטוי גנים הוא זיהוי צבירי גנים בעלי תבנית ביטוי דומה. בעבודה זו פיתחנו שיטה סטטיסטית להערכת חלוקה נתונה לצבירים על בסיס ידע ביולוגי קודם. בעזרת השיטה, ניתן להשוות פתרונות חלוקה שונים, או לברור את הפרמטרים המיטביים עבור אלגוריתמי חלוקה לצבירים. השיטה מבוססת על הטלה למימד יחיד של וקטור הנתונים הביולוגיים אודות כל אחד מן הגנים, במטרה למקסם את היחס בין השונות בין הצבירים ובתוך הצבירים. הציון מחושב על המידע לאחר שהועבר למימד יחיד, על ידי ניתוח א-פרמטרי של השונות. בבדיקות סימולציה שערכנו, הראינו שהשיטה נותנת ציון מדויק יותר מציונים הניתנים בשיטות אחרות. השתמשנו בשיטה שפיתחנו להערכת מספר אלגוריתמי חלוקה לצבירים על בסיס נתונים אודות מחזור התא בשמרים.

2. Chain functions and scoring functions in genetic networks.
Irit Gat-Viks and Ron Shamir.
Published in *Bioinformatics journal supplement for the proceedings of The 11th Annual International Conference on Intelligent Systems for MolecularBiology (ISMB 2003)* [2].

בעבודה זו פיתחנו שיטות בכדי להתמודד עם שתי בעיות בסיסיות בבניית מודלים מתמטיים בעלי יכולת ניבוי עבור רשתות שעתוק. הבעיה הראשונה היא בחירת קבוצה רלוונטית אך מצומצמת של פונקציות בקרת שעתוק אפשריות. הסוגיה השנייה הינה בחירת פונקצית הערכה הולמת על מנת לאמוד את טיב פונקציות בקרת השעתוק. הצענו מחלקה מצומצמת של פונקציות בקרה שכיחות במערכות ביולוגיות, הקרויות פונקציות שרשרת. ניתחנו את מספר הפונקציות במחלקה והראינו שהוא קטן אקספוננציאלית ממספר פונקציות הבקרה הבוליאניות עם אותו מספר של גורמי שעתוק. הגדרנו שתי פונקציות הערכה חדשות המבוססות על שיטות סטטיסטיות מוכרות: אחת אומדת את ההתאמה בין קבוצת גורמי השעתוק וגן המטרה, והשנייה מטרתה הערכה של פונקציה ספציפית לשליטה בשעתוק. בבדיקת האלגוריתם על נתונים ביולוגיים ממערכת ייצור הגלקטוז בשמרים, הראינו את היתרונות שיש לשימוש בפונקציות השרשרת ובשיטות ההערכה שפיתחנו, ואת הדיוק ששיטתנו מקנה בבניית רשתות בקרת שעתוק.

3. Reconstructing chain functions in genetic networks.

Irit Gat-Viks, Roded Sharan, Richard M. Karp and Ron Shamir.

Published in P*roceedings of the Pacific Symposium on Biocomputng (PSB 04)* [3] and in *SIAM Journal of Discrete Mathematics* [4].

מאמר זה מבוסס על הפרדיגמה של פונקציות שרשרת שפותחה במאמר הקודם. במאמר זה חקרנו את הבעיה החישובית של שחזור פונקציות שרשרת תוך שימוש במספר מזערי של ניסויים, כשבכל אחד מהם יש צורך במספר מועט של התערבויות גנטיות. התמודדנו עם הבעיות של מציאת קבוצת הרגולאטורים של פונקציות השרשרת, ועם בעיית שחזור פונקציות הבקרה בהינתן קבוצת הרגולאטורים. הצענו תכניות מיטביות לשחזור פונקציות השרשרת עבור מספר מצבים אפשריים, והדגמנו אותן על נתונים ביולוגיים. המחקר התמקד בסבוכיות התיאורטית של השחזור, תחת הנחה שהמדידות הינן מדויקות ושניתן לשחזר את הפונקציות באופן נפרד משאר הרשת.

4. Modeling and analysis of heterogeneous regulation in biological networks.

Irit Gat-Viks, Amos Tanay and Ron Shamir.

Published in *Proceedings of the First RECOMB Satellite Workshop on Regulatory Genomics* [5] and in *Journal of Computational Biology (JCB)* [6].

בעבודה זו פיתחנו מודל לייצוג רשתות מולקולאריות, והצענו אלגוריתמים לתיקון והשלמת פרמטרים במודל על בסיס מדידות ביולוגיות. גישתנו הייתה לבנות מודל התחלתי על בסיס ידע מוקדם, ואז לשפר את הידע על ידי הגדלת ההתאמה בין ניבויי המודל והמדידות הניסיוניות. השיטה מאפשרת שימוש בנתונים מניסויים רחבי היקף שמודדים הרבה מרכיבים מולקולאריים בניסוי אחד, בד בבד עם שימוש במדידות מניסויים המודדים ישויות מולקולאריות ספציפיות. המודל מייצג מגוון מרכיבים מולקולאריים (כגון רנ"א שליח, חלבונים ומטבוליטים) ורמות שונות של יחסי בקרה (כגון בקרת מטבוליזם, בקרת שעתוק, תרגום ועוד), ומבוטא בעזרת פונקציות בקרה דיסקרטיות ודטרמיניסטיות. מחקר זה מציב שתי בעיות אלגוריתמיות: כיצד להשתמש במודל זה על מנת לחזות את מצבם של משתנים חבויים (מרכיבים מולקולאריים שלא נמדדו בניסוי), וכיצד לעדן ולהרחיב את המודל. הראינו שבעיות אלו הן קשות במקרה שקיימים היזונים חוזרים במערכת, והצענו שיטות לפתרון הבעיות. המתודולוגיה הופעלה על מודל מסלול הביוסינטזה של ליזין בשמר, על בסיס מדידות של ביטוי גנים, כמויות חלבונים וקצבי גידול. בבדיקת האלגוריתם ללימוד פונקציות הבקרה, נצפה שיפור משמעותי בדיוק לעומת אלגוריתמים אחרים שפורסמו במאמרים קודמים.

5. A probabilistic methodology for integrating knowledge and experiments on biological networks.

Irit Gat-Viks, Amos Tanay, Daniella Raijman and Ron Shamir.

Published *in Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 05)* [7] and in *Journal of Computational Biology (JCB)* [8].

מאמר זה הוא למעשה הכללה של המאמר הקודם, המחליף את המודל הקומבינטורי במודל הסתברותי. בעבודה זו פיתחנו מסגרת חישובית רחבה הכוללת פורמליזציה של ידע איכותי, מידול הסתברותי, ומיזוג של נתונים ניסיוניים בהיקף נרחב. שיטותינו מאפשרות לפרש את המדידות בהקשר של הידע האיכותי המוקדם על המערכת, לייחס משמעות סטטיסטית למידת הביטחון בידע המוקדם, וללמוד מודלים משופרים עם התאמה טובה יותר לניסויים. הייצוג הוא על מודל גרפי הסתברותי המאפשר ניתוח של מדידות חלקיות והכללת היזונים חוזרים. בדקנו את ביצועיהם של מספר אלגוריתמי חיזוי והראינו שניתן לחזות בדיוק גבוה משתנים חבויים. בנוסף, פיתחנו אלגוריתמים המבוססים על בחינת השערות סטטיסטית על מנת לאמוד את המשמעות של פונקציות הבקרה הנלמדות. השתמשנו בשיטות אלו ללמוד את פונקציות בקרה שלא אופיינו עד כה בתגובה של שמרים ללחץ אוסמוטי.

6. Refinement and expansion of signaling pathways: the osmotic response network in yeast.

Irit Gat-Viks and Ron Shamir.

To be published in *Genome Research* [9].

במאמר זה המשכנו לפתח את מתודולוגית המידול והניתוח שהוצגה במאמר הקודם. אנו משתמשים באותו מודל המייצג את הידע הקיים על המערכת, ומציעים שיפורים על בסיס ההתאמה לניסויים. בנוסף לשינויים הלוגיים שהוצעו במאמר הקודם, כאן אנו מציעים שיפורים *מבניים* למודל. פיתחנו אלגוריתמים לשיפור היחסים בין רכיבי המודל, ולהרחבה של המודל כך שיכלול רכיבים נוספים המבוקרים על ידי רכיבים מהמודל המקורי. בעזרת מתודולוגית המידול שבידינו, ייצגנו את הידע הקיים על ארבעה מסלולי הולכת סיגנל הקשורים בתגובה ללחץ אוסמוטי בשמר. על בסיס ניתוח של מעל 100 פרופילי רמות שעתוק, האלגוריתם זיהה שלושה יחסים חדשים ברשת. כמו כן, האלגוריתם ניבא קבוצות גדולות של גנים בעלי פונקציות בקרת שעתוק משותפות, המרחיבות את הרשת המקורית. מניתוח קבוצות הגנים ופונקציות הבקרה שהתקבלו, גילינו שמספר גורמי שעתוק ואנזימי מפתח שדרך פעולתם נחקרה רבות בעבר, משפיעים ככל הנראה באופנים חדשים שלא היו ידועים עד כה.