The factor graph network model for biological systems

Irit Gat-Viks, Amos Tanay, Daniela Raijman, and Ron Shamir

School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel.
Email: iritg@tau.ac.il,amos@tau.ac.il,rshamir@tau.ac.il.

Abstract. We introduce an extended computational framework for studying biological systems. Our approach combines formalization of existing qualitative models that are in wide but informal use today, with 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 develop methods for inference and learning in the model. We compare the performance of standard inference algorithms and tailor-made ones and show that hidden variables can be reliably inferred even in the presence of feedback loops and complex logic. We develop a formulation for the learning problem in our model which is based on deterministic hypothesis testing, and show how to derive p-values for learned model features. We test our methodology and algorithms on both simulated and real yeast data. In particular, we use our method to study the response of *S. cerevisiae* to hyper-osmotic shock, and explore uncharacterized logical relations between important regulators in the system.

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 amounts of information on the genome-wide behavior of the system. Genome-wide datasets are subject to extensive computational analysis, but their integration into existing (informal) 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 denovo reconstruction of a network on genes (e.g., [6]), used prior knowledge on network topology (e.g., [9, 11]), or combined transcription factor location and sequence data to learn a clustered model for the genome-wide behavior of the system [1, 23, 2]. Other studies built detailed models manually, utilizing existing biological knowledge [3, 5] but lacked computational methods for model reassessment in light of additional evidence.

In this study we describe a new algorithmic 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 [25, 8] 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 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 previous works that modeled gene networks as Bayesian networks (or dynamic Bayesian networks) on genes, our model expresses diverse biological entities (e.g., mRNAs, proteins, metabolites), and directly accommodates undelayed feedback loops which are essential in many biological systems. We formalize our model as a probabilistic factor graph [14], and show how it can be developed naturally from basic assumptions on the biochemical processes underlying the regulatory network and on the information we have on it.

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 between 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. For lack of space some proofs and details are omitted.

2 Modeling prior knowledge and experimental observations

In this section we present our probabilistic model for a biological regulatory network. 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 a 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. Regulation functions approximate an underlying biochemical reaction whose exact parameterization is not known. Note that the regulatory process is stochastic at the single cell level, but the parameters of the reaction equations governing it are deterministic. Moreover, 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 strategy may assume deterministic logic and noisy observations. Given these notions, we wish to find a discrete, deterministic model that adequately approximates the system's reaction equations, in a way that can take into account the different sources of noise and uncertainty in the data.

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 determine the regulation function with high probability if our prior knowledge is very strong. At the other extreme end, lack of information is modeled by uniform distribution over all possible regulation functions.

We formalize these notions as follows (see Figure 1). Let $X = \{X_1, ...X_n\}$ be a collection of biological variables. Let $S = \{0, 1, ..., k-1\}$ 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 $Pa_i = \{Pa_{i,1}, ..., Pa_{i,d_i}\} \subseteq X$. 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 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 as in standard Bayesian networks, by the conditional probabilities θ^i:

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

Note that in current applications, the interpretation of the θ distributions is not as representing a stochastic process in which θ^i defines the conditional probability of X_i given its regulators. 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. In the future, given refined understanding of regulatory switches, and measurements at the single cell level, the θ distributions may be applicable to describe the inherent stochasticity of some biological switches.

We wish to learn regulation functions from experimental data. 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* $Y_1, .., Y_n$ 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 Gaussians (Figure 1b) to model ψ^i, but other formulations are also possible.

Our model is now represented as a probabilistic factor graph [14], defined

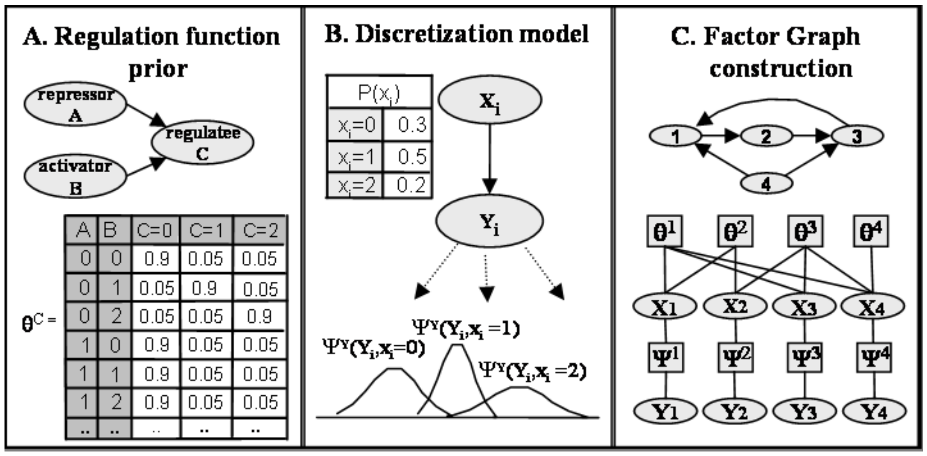
by the joint distribution over logical (X) and sensor (Y) variables:

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

Where Z is a normalization constant. We call this formulation a *factor graph network (FGN) model*. Factor graphs are widely used probabilistic graphical models that were originally applied to coding/decoding problems. Recently, factor graphs were used in computational biology, although in a different context [27].

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. [18]). 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 equals the original beliefs $\theta(X_i, Pa_i)$. We note that derivation of the FGN model from basic assumptions on deterministic approximations of the biological system and on our prior beliefs on them is possible, and will be described in a later publication.

We now outline several important extensions of our model that are not used in this study. In its simplest form, our model describes the steady state behavior of the system. Biological processes are ideally described as temporal processes, but when sampling rate is slow relative to the rate of the regulatory mechanisms, the steady state assumption can be invoked. Different regulatory processes operate on different time scales: In the typical high throughput experimental sampling rate, the steady state assumption may be highly adequate for metabolic pathways and post translational regulation and reasonable for transcriptional programs. For the models considered in this work we have combined interactions from all types and validated empirically (using, e.g., cross validation) that the steady state assumption still enables biologically meaningful results. Our model is unique in its handling of steady state feedback loops. It can be extended to handle slower temporal processes in a way analogous to the construction of dynamic Bayesian networks (DBN) [7, 24] 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 possible extension is the consideration of other classes of regulation functions (for example, we can consider continuous or ranked function as in [26, 22, 12, 16]).



$\theta^C =$

A	B	C=0	C=1	C=2
0	0	0.9	0.05	0.05
0	1	0.05	0.9	0.05
0	2	0.05	0.05	0.9
1	0	0.9	0.05	0.05
1	1	0.9	0.05	0.05
1	2	0.9	0.05	0.05
..	..	..	..	..

P(x_i)	
x_i=0	0.3
x_i=1	0.5
x_i=2	0.2

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 is transformed into a factor graph, using the regulation function priors and the discretizers' distributions

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 (Y) variables are observed in each experiment (for example, mRNA variables are determined in a gene expression experiment). The values of some logical (X) variables may also be determined by the experimenter (e.g., the nutritional condition is represented by the states of extra-cellular metabolite variables). Perturbations of certain regulation functions (e.g., by gene knockouts) are not discussed here for simplicity, but are easily incorporated by modifying the θ parameters at the appropriate conditions. The inference problem is defined with respect to an observation on a subset of the variables. The goal is to compute the posterior distribution of the unobserved (hidden) variables, i.e. the probability distribution of hidden variables values given the model and the observed data. For example, given a gene expression profile, we can compute the posterior distribution of protein variables or estimate the level of a certain metabolic intermediate. Solving instances of the inference problem is a pre-requisite for learning refined models.

Inference in graphical models is an NP hard problem [4] that was extensively studied. A popular class of algorithms [28] uses an approximation of the posterior distribution assuming certain decomposition over independent variables or clusters of variables. Algorithms from this class include loopy belief propagation (LBP), mean field, and their generalizations. They provide in many cases an effective way for estimating posteriors of small sets of variables. A different approach is needed if one is interested in *posterior modes* - complete system states with high probability mass. Modes provide an important insight into the system's state space, and cannot be directly computed from independent variable posteriors.

Posterior modes are particularly important when we study the behavior of systems with feedback loops. For example, the single variable posteriors in a positive feedback loop may contain no information even though the loop has two simple modes (all variables on or all variables off) that absorb all the probability mass. Still, when applicable, algorithms that compute independent posteriors perform well.

We studied the effects of our model's specific characteristics on the performance of several inference algorithms. We implemented a Gibbs sampler, the LBP algorithm, and a modes-based instantiation inference algorithm. Our **Gibbs sampler** is a naive MCMC algorithm [15] that performs a random walk over the space of model states, based on sampling from local distributions. In our model, sampling is done only for the X variables (unobserved sensors do not affect other posterior distributions). In Gibbs sampling, the posterior is estimated from the set of terminal trajectory states, so any query from the posterior is in principle possible (including single variable posteriors or complete modes). The **LBP** algorithm for the FGN model was implemented as described in [28]. The algorithm is a message passing procedure that is guaranteed to reach the exact solution for acyclic models. The algorithm, like others in the family, decomposes the posterior across variables or sets of variables, so we could only use it to obtain the posterior of small sets of variables. We also developed an instantiation-based inference algorithm that exploits the known dependency structure of the model and builds on ideas from [8]. The **modes instantiation (MI)** algorithm first builds a deterministic 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 (see [8]) and enumerate all possible value assignments for variables in this set. Each feedback set configuration gives rise to a unique model state, which is used as the basis for further greedy optimization of the state likelihood. We add to the resulting set of high probability modes an additional small set of states derived using the Gibbs sampler. We now estimate the posterior as a mixture of the set of locally optimal and sampled model states, weighted by their likelihoods, where the partition function (our estimation of the Z parameter) equals the sum of likelihood over all of the states we consider. We thus map the posterior function landscape using a limited set of local optima.

We tested the three inference algorithms on a simulated model (see Figure 2). We constructed simulated FGN models by adaptation of a deterministic model. 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 detailed description of the simulation, see our website (www.cs.tau.ac.il/$\sim$rshamir/fgn/). 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 one represent very good knowledge on the system under study. Models with α near $\frac{1}{k}$ represent complete lack of knowledge. Our analysis (see Figure 2) indicates that for estimation of single variable posteriors, LBP outperforms the other two algorithms (and also the mean field algorithm and a simple clustered variational algorithm [13], data not shown). Also, when the prior is strong, MI provides reasonable accuracy. However, the distribution

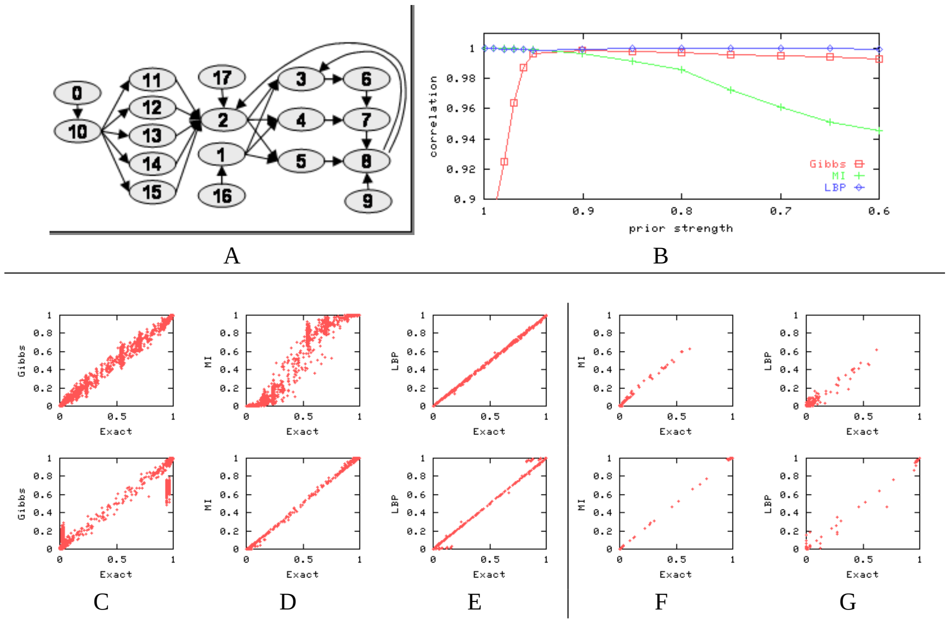


Fig. 2. Performance of different inference algorithms on a simulated model. A) The dependency graph G_R of the simulated model. B) Effect of prior strength on inference accuracy. Y axis: the correlation of approximated and exact single variable posteriors. X axis: prior strength (α). For strong priors, 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 modes is a poor approximation for the posterior in these cases. C,D,E) Detailed correlation of algorithmic and exact single variable posteriors for $\alpha = 0.7$ (top) and $\alpha = 0.97$ (bottom). F,G)Detailed correlation of algorithmic 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 modes.

of posterior modes produced by MI is more accurate than that of LBP (Figure 2F,G), exemplifying the limitations of the posterior independence assumptions. Overall, we prefer using LBP to infer single variable posteriors and MI to approximate the global 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 accumulated biological knowledge. There are several alternative approaches to discretization. In most previous works on discrete models (e.g., [6, 8]), discretization was done as a preprocess, using some heuristic rule to map real valued measurements into discrete states. In these cases, the rule must be determined and tuned rather arbitrarily, and typically, all variables are discretized using the same rule. The FGN model suggests a different approach to discretization. Here the discretization is an integral part of the model, so the dependencies between the discretization scheme 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 start by initializing all ψ using any heuristic discretization scheme. On 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 over-fitting and may decrease learnability. One can control such undesired effects by using the same 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. The standard approach to learning in graphical models seeks parameters that maximize a likelihood or Bayesian score, possibly completing missing data using an EM algorithm. The applicability of this approach relies heavily on our ability to interpret the learned parameters in meaningful way. In many cases such interpretation is straightforward (e.g., learning the probabilities of each face in an unfair dice). In other cases, for example, when learning regulation functions in the FGN model, such interpretation is less obvious and should be carefully evaluated given the phenomenon we try to model. Specifically, if we assume the parameters of the logical factors in the FGN model represent our prior beliefs on the logical relations between variables, we may attempt to learn by confirming beliefs (deciding if a certain regulator assignment gives rise to a certain regulatee assignment) rather than finding optimal new beliefs. Given that currently most biological systems are only roughly characterized, and available experimental data provide estimated averages of the states of biological factors in them, this hypothesis driven approach may be a realistic alternative to parameters learning. In other cases (most importantly, when single cell measurements are available), we may prefer the standard approach. We assume throughout that the network topology is fixed and focus on learning regulation functions only. In principle, the methods introduced below can also be used to refine the topology (add and remove edges).

We focus on the regulation of some variable 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 $j' \neq j$. 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 each h_j given the data, essentially solving the inference problem in the h_j model to compute the full probability of the data. Computing the full probability is a difficult problem, and as stated in section 3 we approximate it using the MI algorithm. We bound the likelihood of the data given h_j by summing the likelihoods of the posterior modes sampled by the MI algorithm (including modes computed from configurations of a feedback set in G_R

and modes obtained using Gibbs sampling). While this is a very crude approximation, our empirical analysis shows it is still adequate (see below).

To assign statistical meaning to the learning procedure we use two methods: bootstrap and likelihood ratio testing. In the bootstrap method, we re-sample conditions from the original data D and reiterate 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.

We fix j and define $H_1 : h_j, H_0 : \cup_{k \neq j} h_k$ and $\lambda = \frac{max_{h_i \in H_0 \cup H_1} Pr(D|h_i)}{max_{h_i \in H_0} Pr(D|h_i)}$. In order to compute a p-value for the observed statistic λ, we have to estimate the distribution of λ given H_0. To that end we generate samples from the distribution $Pr(D|H_0)$, compute the corresponding λ's and reconstruct the distribution. When sampling datasets D, we take into account both the model (defined by H_0) and the properties of the original dataset. We do this as follows: for each of the conditions in the original dataset, we fix all observations of X variables (these correspond to the experimental conditions) and all model perturbations. We then apply the MI algorithm with the H_0 model, and compute the set of posterior modes matching these experimental conditions, without using any of the observations on Y variables. We then generate a sample by a) selecting a mode from the mixture, and b) generating observations on Y variables using the model discretizer distributions ψ.

We analyzed the performance of the bootstrap and likelihood ratio test methods by learning features in a simulated model (see our website 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 global posterior, it is less 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 lysine intake and biosynthesis pathway, and the Hog1 MAPK pathway, which mediates the yeast response to osmotic stress. For each of the models, we performed extensive literature survey in order to construct the initial model of the system (for the lysine system, our previously developed deterministic model [8] was the main source). The HOG model is outlined in Figure 6. The lysine model contained 140 variables and the HOG model contained 50 variables. We used prior strength $\alpha = 0.9$ in all reported experiments. We collected published experimental data on each of the models. The data consisted of 23 conditions (cf. [8]) for the lysine model and 129 conditions for the HOG model [17]. Differential measurements from cDNA microarrays were transformed into absolute values as described in [8].

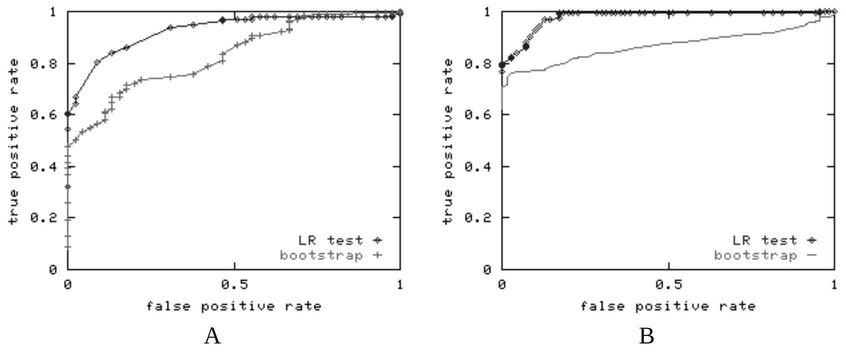


Fig. 3. Accuracy of learning regulation functions. Each figure is a ROC curve (X-axis: false positives rate Y-axis: true positives rate) for learning the functions in a simulated model using bootstrap and likelihood ratio test for determining the significance of learned features. Results are shown for learning from 15 (A) and 80 (B) conditions. The accuracy of the likelihood ratio test method is consistently higher.

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.

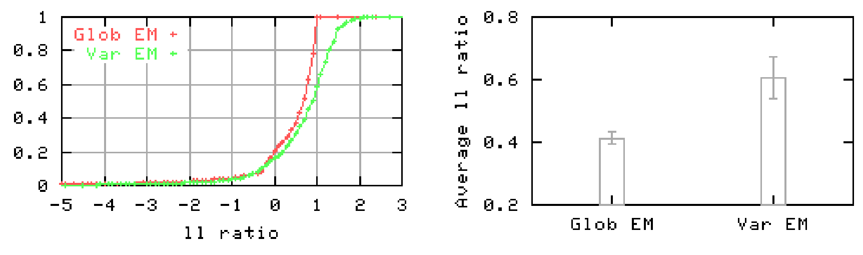
We used cross validation to compare three alternatives: (A) single common predefined mixture of Gaussians. (B) using the EM algorithm described in Section 4 to learn a single common maximum likelihood ψ distribution (C) applying an unconstrained EM to learn variable specific ψ^i-s.

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 4 shows the results of the cross validation on the HOG model. 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: likelihood ratios smaller than 0 represent cases where the refined discretization resulted in overfitting, ratios larger than 0 represent successful generalizations. We conclude that for 80% of the cases, incorporating the discretization into the model significantly improves performance. Moreover, variable specific discretization outperforms the optimized global discretization scheme. Similar results were obtained for the lysine model.

6.2 Learning Regulation functions

We used cross validation in the lysine model to confirm the capability of our method to learn real regulation functions from real data and to compare its performance to the deterministic and to a naive Bayesian approaches. The deterministic model approach [8]

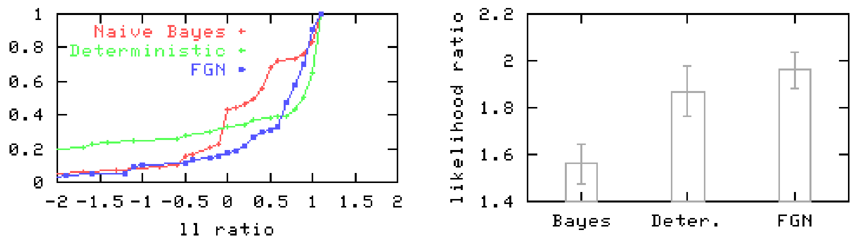


A B

Fig. 4. Learning discretization distributions. Cross validation results for alternative methods for estimating the discretization functions ψ^i in the HOG model. Glob EM - optimized single common discretization function. Var EM - optimized variable specific discretization. A) Cumulative distribution of log likelihood (ll) ratios comparing each of the two discretization methods to the global preprocessed discretization scheme. B) Average ll ratios for the two methods. Bars indicate the predicted standard deviation of the averages.

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 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 regulators assignment. 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 prior for the feature ($\alpha = 0.99$) in case a decision was made, and a uniform distribution prior 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 5 depicts the log likelihood ratio distribution for the three methods (compared to a uniform 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 does its deterministic counterpart, probably due to its ability to tolerate mistakes in the prior model. Both the deterministic and probabilistic methods make good use of the additional knowledge,



A B

Fig. 5. **Learning regulation functions.** 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 [8], and learning using the FGN model. Bars indicate the predicted standard deviation of the averages.

formalized into the model logic, to obtain better results than the naive, topology based approach.

6.3 Biological analysis of the HOG model

The response of yeast to hyper-osmotic stress is mediated through parallel MAPK signaling pathways, the multi target 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 [10]. We have constructed an FGN model that represents known regulatory relations in the HOG system (Figure 6A) and used it to study the transcriptional program following treatment by variable levels of KCl [17]. The data we used contained observations of all mRNA variables in the model and assignments of fixed values for logical variables describing experimental conditions (e.g., Turgor pressure). We used the MI inference algorithm to estimate the states of all logical variables in the model and applied the model to test the accuracy of the prior logic distributions modeled from the literature. We summarize the model predictions in the *discrepancy matrix* shown in Figure 6B. The discrepancy matrix shows the correspondence between model predictions and experimental observations at the level of a single variable under a single 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 posterior distribution of Y_i given the observations on all other variables and compute the expected value and the probability of Y_i observation. We present the difference between the predicted values and the observations in a color coded matrix.

The discrepancy matrix reveals several important discrepancies between the current model for osmo-regulation 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 the osmoregulation program (10-30 minutes after treatment with 0.5M KCl, around 60 minutes in the 1M KCl treatment). This repression is not reported in the current literature. We hypothesize

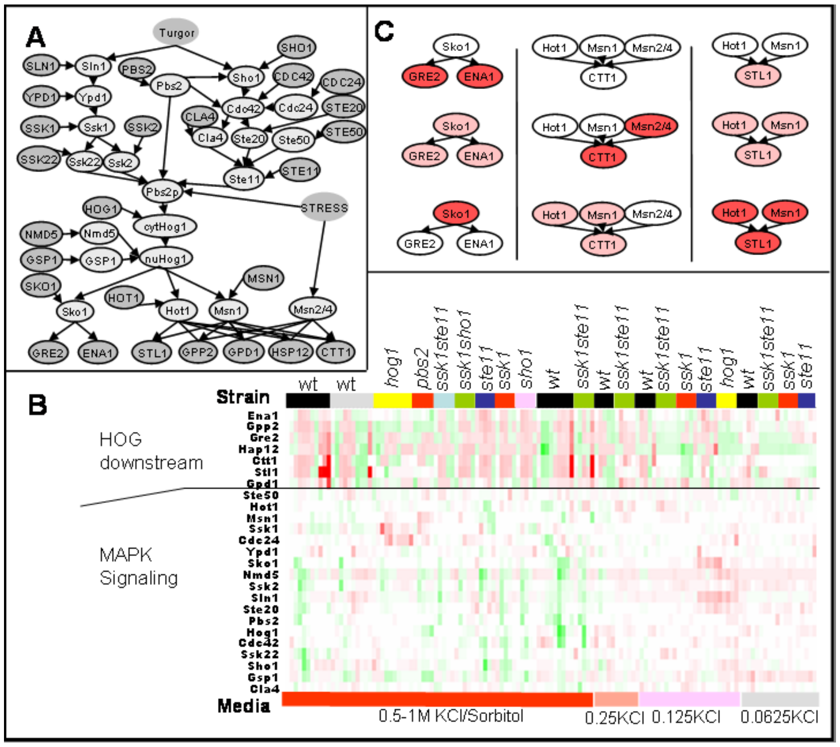


Fig. 6. **Analyzing yeast response to hyper-osmotic shock.** mRNA variable names are capitalized, protein variables names appear in initial capital letters. Stimulator variables appear as unoutlined ovals. A) Topology of the HOG FGN model used in this study. B) The discrepancy matrix for the HOG model and data from O'Rourke et al. Columns correspond to mRNA variables and rows to experimental conditions, see text for details. Green (Red) cells indicate observations that are smaller (bigger) than the expected prediction. Color intensity is proportional to minus the log likelihood of the observation given the inferred variable posteriors. C) 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 as node colors: white - 0, pink - 1, red - 2.

that as part of the adaptation to high levels of osmotic pressure, yeasts may reduce the Hog1 signaling cascade sensitivity, 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 regulators Sko1, Hot1, Msn1 and Msn2,4 [19, 21, 20]. 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 thus 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 6c shows examples of logical relations we learned. First, we were able to learn the known repressive role of Sko1 in the regulation of *GRE2* and *ENA1* [19]. We learned three model features that associated the mRNA variables of these two genes with the opposite state of the inferred Sko1 regulator state. Note that the expression of the *SKO1* gene during osmotic stress is static, and the correct regulation function could only be learned given the inferred Sko1 activities. These inferred activities take into account, in addition to the mRNA measurements, the entire model and its regulatory functions. A second example for a variable we learned regulation for was *STL1*. The regulation of *STL1* is reported to be completely dependent on Hot1 and Msn1 [20], but it is not clear what are 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 (we model just four representatives of a larger regulon: *GPP2*, *GPD1*, *HSP12* and *CTT1* [21]). Our results indicate that the two signaling pathways (the HOG cascade and the general stress pathway) act in parallel, and each of the branches can induce the regulon in the absence of activity from the other.

Acknowledgment - We thank Nir Friedman, Dana Pe'er and the anonymous referees for helpful comments. IGV was supported by a Colton fellowship. AT was supported in part by a scholarship in Complexity Science from the Yeshaia Horvitz Association. DR was supported by a summer student fellowship from the Weizmann Institute of Science. RS 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

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. M.A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117:185–198, 2004.
3. K.C. Chen et al. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol Biol Cell*, 11:369–91, 2000.
4. G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
5. M.W. Covert, E.M. Knight, J.L. Reed, M.J. Herrgard, and B.O. Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429:92–96, 2004.
6. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comp. Biol.*, 7:601–620, 2000.
7. N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *Proc. 14th Conference on Uncertainty in Artificial Intelligence.*, pages 139–147, 1998.
8. I. Gat-Viks, A. Tanay, and R. Shamir. Modeling and analysis of heterogeneous regulation in biological networks. In *Proceedings of the first Annual RECOMB Satellite Workshop on Regulatory Genomics*, pages 21–31, 2004. Also J. Comput Biol in press.
9. A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Combining location and expression data for principled discovery of genetic regulatory networks. In *Proceedings of the 2002 Pacific Symposioum in Biocomputing (PSB 02)*, pages 437–449, 2002.

10. S Hohmann. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol Mol Biol Rev.*, 66(2):300–72, 2002.
11. S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *J. Bioinform. Comput. Biol.*, 2:77–98, 2004.
12. S. Imoto, S. Kim, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *J. Bioinform. Comput. Biol.*, 1:231–252, 2004.
13. T.S. Jaakkola. Tutorial on variational approximation methods. In D. Saad and M. Opper, editors, *Advanced Mean Field Methods - Theory and Practice*, pages 129–160. MIT Press, 2001.
14. F.R. Kschischang, B.J. Frey, and Loeliger H. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.
15. D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 175–204. Kluwer Academic Press, 1998.
16. I. Nachman, A. Regev, and N. Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20:248–256, 2004.
17. S.M. O'Rourke and I. Herskowitz. Unique and redundant roles for hog mapk pathway components as revealed by whole-genome expression analysis. *Mol Biol Cell.*, 15(2):532–42, 2004.
18. J. Pearl. *Probabilistic Reasoning in intelligent systems*. Morgan Kaufmann publishers, inc, 1988.
19. M. Proft and R. Serrano. 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.*, 19:537–46, 1999.
20. M. Rep, M. Krantz, J.M. Thevelein, and S. Hohmann. 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, 2000.
21. M. Rep, V. Reiser, U. Holzmller, J.M. Thevelein, S. Hohmann, G. Ammerer, and H. Ruis. Osmotic stress-induced gene expression in saccharomyces cerevisiae requires msn1p and the novel nuclear factor hot1p. *Mol. Cell. Biol*, 19:5474–5485, 1999.
22. M. Ronen, R. Rosenberg, B. Shraiman, and U. Alon. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proceedings of the National Academy of Science USA*, 99:10555–10560, 2002.
23. 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.
24. V.A. Smith, E.D. Jarvis, and A.J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18:216–224, 2002.
25. A. Tanay and R. Shamir. Computational expansion of genetic networks. *Bioinformatics*, 17:S270–S278, 2001.
26. A. Tanay and R. Shamir. Modeling transcription programs: inferring binding site activity and dose-response model optimization. *J. Comp. Biol.*, 11:357 – 375, 2004.
27. C.H. Yeang, T. Ideker, and T. Jaakkola. Physical network models. *J Comput Biol.*, 11(2-3):243–62, 2004.
28. S. Yedidia, WT. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2004-040, Mitsubishi electric resaerch laboratories, 2004.