Bioinformatics, 2020, 1–8
doi: 10.1093/bioinformatics/btaa122
Advance Access Publication Date: 25 February 2020
Original Paper

Systems biology

A probabilistic graphical model for system-wide analysis of gene regulatory networks

Stephen Kotiang and Ali Eslami*

Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on June 28, 2019; revised on January 15, 2020; editorial decision on February 15, 2020; accepted on February 18, 2020

Abstract

Motivation: The inference of gene regulatory networks (GRNs) from DNA microarray measurements forms a core element of systems biology-based phenotyping. In the recent past, numerous computational methodologies have been formalized to enable the deduction of reliable and testable predictions in today's biology. However, little focus has been aimed at quantifying how well existing state-of-the-art GRNs correspond to measured gene-expression profiles.

Results: Here, we present a computational framework that combines the formulation of probabilistic graphical modeling, standard statistical estimation, and integration of high-throughput biological data to explore the global behavior of biological systems and the global consistency between experimentally verified GRNs and corresponding large microarray compendium data. The model is represented as a probabilistic bipartite graph, which can handle highly complex network systems and accommodates partial measurements of diverse biological entities, e.g. messengerRNAs, proteins, metabolites and various stimulators participating in regulatory networks. This method was tested on microarray expression data from the M^{3D} database, corresponding to sub-networks on one of the best researched model organisms, *Escherichia coli*. Results show a surprisingly high correlation between the observed states and the inferred system's behavior under various experimental conditions.

Availability and implementation: Processed data and software implementation using Matlab are freely available at https://github.com/kotiang54/PgmGRNs. Full dataset available from the M^{3D} database.

Contact: ali.eslami@wichita.edu

1 Introduction

Modeling the coupled dynamics of gene (protein) expression patterns in accordance with changing internal and environmental conditions is an important task in systems biology. To characterize and uncover the exact dynamics of genome-wide gene regulatory networks (GRNs), significant research effort has been devoted to continuously refining computational methods that will allow researchers to understand the complex interactions of gene regulations (Hughes *et al.*, 2000). Such methods, often referred to as reverse engineering (Karlebach and Shamir, 2008; Madhamshettiwar *et al.*, 2012; Prill *et al.*, 2010; Stolovitzky *et al.*, 2007), have been used to fit discrete models of GRNs to high-throughput experimental data. In the literature, gene expression-based inference approaches have shown modest performance when applied to real data compared to *in silico* expression data (Madhamshettiwar *et al.*, 2012; Marbach *et al.*, 2012). In addition, predictive performance over a purely microarray expression-based approach can be improved by incorporating multiple types of data, such as gene set enrichment (Chouvardas *et al.*, 2016), sequence information (Yu *et al.*, 2014) and network topology (Hartemink *et al.*, 2001).

On the other hand, GRNs have commonly been modeled using ordinary differential equations (ODE), Boolean networks and probabilistic graphical models including Bayesian networks (de Hoon *et al.*, 2002; Friedman *et al.*, 2000; Lovrics *et al.*, 2014). For the reconstructed GRN model reassessment in light of additional evidence, in the recent past, computational methodologies have been developed and formalized mathematically, in order to rigorously integrate prior biological knowledge and high-throughput measurements (Covert *et al.*, 2004; Gat-Viks *et al.*, 2006). Furthermore, such methodologies have been formalized in a manner that allows for good predictive descriptions of experimental data. Regardless of the modeling or computational approach applied, it is important to assess the validity of such networks. Given the topology of a biological network and a partial set of microarray expression profiles for all genes in the network, a reverse engineering algorithm must infer a probabilistic dynamical system that best *explains* the observed experimental data. In this article, we consider this reverse engineering problem. We describe the *dynamics* of a network as trajectories of gene-expression levels at steady state, given experimental conditions.

In the literature, some methods that can take a biological network and simulate biological data of different genes as either time-series data or steady-state values have been proposed. One of these is *sgnesR* (Tripathi *et al.*, 2017), an R package used to simulate a gene-expression profile from a given gene network using the stochastic simulation algorithm, for which the reaction parameters are specified under defined constraints. Similarly, a multi-view genomic data simulator proposed by Fratello *et al.* (2015) can generate synthetic biological data from ODE-based network models with known parameters, constructed through an iterative procedure. Simulated datasets, although fully controlled, are often too simplistic to efficiently explain the complex regulatory interactions among biological entities compared to real gene-expression data. Another widely used simulation and modeling tool in systems biology is the complex pathway simulator (COPASI) (Hoops *et al.*, 2006; Klipp *et al.*, 2008). COPASI is a stand-alone program that specializes in setting up and analyzing biochemical and kinetic network models while also providing some basic stoichiometric analyses. It allows for more detailed and fine-grained analysis, but also demands more knowledge, namely about the kinetics of individual processes. An important factor in the simulation of these models is the knowledge of kinetic reaction parameters. This information can be extracted from the literature; however, it is hard to find (Klipp *et al.*, 2008). Lack of kinetic constants stem from difficulty in measurements and uncertainties in the function of many proteins and their interactions, and thus limit the application of some of these approaches. However, these simulators provide valuable information that can be used to test network inference methods qualitatively, as well as to identify model parameters.

In our work, we apply a probabilistic model to statistically assess the global consistency between GRNs and the gene-expression profile of diverse experimental conditions. Therefore, we explore a probabilistic framework that allows us to model uncertainty in cellular networks through integration of prior biological knowledge and high-throughput experimental data. We formalize the model as a probabilistic factor graph (Kschischang *et al.*, 2001), which can handle highly complex systems and extensive datasets. This probabilistic model allows us to overcome the drawbacks of models that assume noiseless observations, because it is able to mix noisy continuous measurements with discrete regulatory relations among variables. Furthermore, it does not require the explicit determination of network kinetic parameters. Our method is applied to *Escherichia coli* DNA microarray data, where it is successfully used to predict the global allowable steady state of genes in the respective extracted sub-networks. Our analyses are performed on real gene-expression data and networks. The method is further validated using network perturbation techniques (Maslov, 2008), as well as gene deletion experiments. The rest of this article is organized as follows: In Section 2, we formulate a probabilistic factor graph network (FGN) framework for the analysis of biological networks given experimental data. We follow on with the inference model by applying message-passing algorithm. Section 3 elucidates examples of the regulatory networks with a brief discussion on data discretization methodology. Section 4 presents statistical analyses of cellular network examples using the described framework. The article is concluded in Section 5.

2 Model and methods

2.1 Probabilistic model for GRN

To analyze the behavior of gene networks, we need to study regulation functions and interactions among biological entities. Such relations, for instance, determine the level of gene expression for a particular gene from a set of transcription factors interacting with the gene. In practice, the information available for regulation mechanisms is incomplete and of variable certainty, which motivates the employment of a probabilistic framework for inferring such functions. In addition, due to the noisy nature of biological experiments, probabilistic models help integrate experimental data in a network context. Therefore, we present a probabilistic model by defining variables and formulating prior biological knowledge on the relations among them. Each biological entity can be modeled as a discrete or continuous random variable. This random variable represents the level in which an entity is present in the cell. Taking partial information into account, we derive a distribution function for each variable that considers interaction relations among them as well as their level of uncertainty.

Figure 1a shows an example of a simple gene network structure having four genes. An unsigned directed edge from gene g_2 to g_1, for instance, means gene g_2 influences the action of g_1, through creation of some specific proteins (Karlebach and Shamir, 2008). Altering the state of one gene can cause other genes interacting with it to change their states, leading to a cascade of modifications in the alleles of genes. The process is repeated iteratively in time until a global steady state is reached. In general, let $g = \{g_1, g_2, \ldots, g_n\}$ denote the set of biological random variables, i.e. genes in a network. In this work, we consider genes as discrete random variables. Each node g_i may take a value from a range of (usually finite) k logical states $s = \{0, 1, \ldots, k-1\}$ that a variable may attain. We denote by $Pa_i = \{Pa_{i,1}, Pa_{i,2}, \ldots, Pa_{i,m}\} \subseteq g$, the set of *parents* of g_i (i.e. set of variables that regulate g_i). By definition, the regulation function f_i for a variable g_i is formulated by the conditional probabilities as

$$f_i \triangleq p(g_i|Pa_i). \qquad (1)$$

This function f_i is referred to as the belief that g_i takes a certain state with respect to an assignment from its parents. If $Pa_i = \varnothing$ (i.e. g_i has no parents), then $p(g_i|\varnothing) = p(g_i)$. We define the learning problem of f_i as selecting the maximum likelihood of data given the model. Formally, the Bayesian network shown in Figure 1a implicitly encodes the joint probability distribution as a product of local conditional distributions:

$$p(g_1, g_2, \ldots, g_n) = \frac{1}{Z}\prod_{i=1}^{n} p(g_i|Pa_i), \qquad (2)$$

where Z is a normalization constant. In this work, we employ a probabilistic factor graph (Kschischang *et al.*, 2001), which explicitly expresses the structure of the joint distribution's factorization in Equation (2). For example, Figure 1b expresses the factorization

$$p(g_1, g_2, g_3, g_4) \;\propto\; p(g_1|g_2)p(g_2|g_3, g_4)p(g_3)p(g_4). \qquad (3)$$

A factor graph visualized as an undirected graph associating variable nodes and factor nodes is defined as a bipartite graph (see Fig. 1b), for instance, of the simple network (Fig. 1a). In this context, a *variable node* denotes each random variable g_i, and a *factor node* denotes a local function f_i. To convert a Bayesian network to a bipartite graph, simply draw an edge between a variable g and a factor f_j, if the scope of f_j contains g. This representation is convenient and has been successfully utilized in the literature to discover new regulatory relationships and even optimize regulation functions (Gat-Viks *et al.*, 2006; Karlebach and Shamir, 2008). However, in practice, biological dependency graph models contain loops, and the situation becomes more complex. In this case, the probabilistic FGN model approximates rather than giving the true belief functions (Yedidia *et al.*, 2005). Furthermore, we note that, given experimental measurements at the single cell level, the regulation functions may suffice to describe the inherent stochasticity of the underlying biochemical reactions.

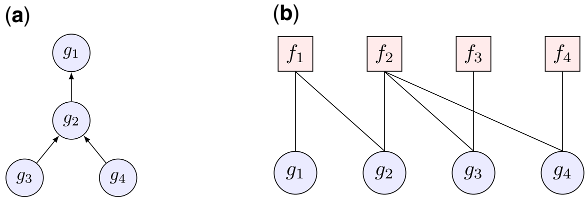

Fig. 1. Graphical probability models: (**a**) Bayesian gene network and (**b**) factor graph equivalent

2.2 Inference in FGN model

In this section, we discuss the problem of probabilistic inference in our FGN model. Typically, each experiment provides partial information on the state of model variables. Therefore, given experimental evidence, e.g. the gene-expression profile D specifying real-value sensor variables and the graphical model, the problem of probabilistic inference seeks computation of the *posterior* distributions for a single hidden (unmeasured) variable. This is referred to as *marginal inference*. Of particular interest, for instance, would be to compute $p(g_i|D)$. Another problem would be to compute the *likelihood* $p(D)$ of the evidence. The probabilistic inference is an NP-hard problem (Cooper, 1990) because it can involve summing an exponentially large number of terms.

In directed acyclic graphs, many exact inference techniques exist in the literature, such as variable elimination, naive brute force marginalization and the family of message-passing algorithms, such as junction tree, sum-product and belief propagation, which perform very well (Aji and McEliece, 2000; Kschischang *et al.*, 2001; Pearl, 2014). However, for loopy graphs (graphs that contain cycles, such as GRNs), messages may circulate indefinitely in loops and hence do not guarantee convergence. Moreover, even if they do, the steady state may not represent marginals of the nodes. Accordingly, sufficient conditions to guarantee uniqueness of fixed points and/or convergence have been studied in the literature (Mooij and Kappen, 2007). In this work, we explore the loopy-belief propagation (LBP) algorithm, a popular message-passing algorithm on loopy graphs. This algorithm belongs to a class of *variational* algorithms, which approximate the marginal-distribution functions, assuming certain decomposition over a cluster of variables or independent variables (Mooij and Kappen, 2007; Murphy *et al.*, 1999; Yedidia *et al.*, 2005). Essentially, it is a fixed-point iterative procedure that tries to minimize the *Bethe* free energy, F_{bethe} (Yedidia *et al.*, 2005). Still it is a well-defined algorithm and empirically often achieves a good approximation if the solution converges (Murphy *et al.*, 1999).

In our FGN model, we use belief propagation as the message-passing protocol. We compute messages sent between 'variable nodes' corresponding to dashed ellipses for the equivalent FGN shown in Figure 2, as functions of the parent p. A message sent from p to child c is denoted as $\mu_c(p)$, while a message sent from c to p is denoted as $\lambda_c(p)$. Note that within an ellipse, the message sent from g_i to the local function f_i is given by the product of all incoming λ messages. Similarly, a message from f_i to g_i is the product of f_i with other μ messages received at f_i summarized for g_i. Formally, denote the set of parents of a gene variable g_i by Pa_i, and the set of children of g_i by C_i. Therefore, for every $a \in Pa_i$,

$$\lambda_{g_i}(a) = \sum_{\sim\{a\}} \left(\prod_{d \in C_i} \lambda_d(g_i) \; p(g_i|Pa_i) \prod_{b \in Pa_i \ \{a\}} \mu_{g_i}(b) \right), \tag{4}$$

and for every $d \in C_i$,

$$\mu_d(g_i) = \prod_{c \in C_i \ \{d\}} \lambda_c(g_i) \sum_{\sim\{g_i\}} \left(p(g_i|Pa_i) \prod_{a \in Pa_i} \mu_{g_i}(a) \right), \tag{5}$$

where $\sum_{\sim\{x\}}$ is the summary operator over all variables except x. Nodes are updated in parallel, and both λ and μ messages are normalized at each iteration. Computation of a marginal function of an individual variable, referred to as the *belief* of a node in the factor graph and denoted by $b_i(\cdot)$, is given by the product of all messages received by the node:

$$b_i(g_i) = \prod_{d \in C_i} \lambda_d(g_i) \sum_{\sim\{g_i\}} \left(p(g_i|Pa_i) \prod_{a \in Pa_i} \mu_{g_i}(a) \right). \tag{6}$$

The message-passing algorithm continues until the solution converges or no significant difference in belief update occurs in the order of 10^{-4}. In all our inferences, the initial messages were set to a uniform vector of ones. However, in certain cases, random initialization yielded similar results since the initial conditions rapidly 'fade out'.

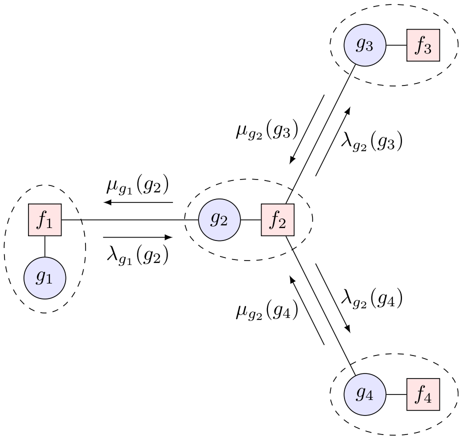

Fig. 2. Messages sent in belief update of FGN example shown in Figure 1b. Arrows indicate flow of belief messages sent between variable nodes (i.e. dashed ellipses corresponding to nodes in Bayesian gene network), μ denotes message sent from parent gene to child gene in FGN model, and conversely, λ represents message from child to parent

3 Network models and data

In this article, we test the applicability of our model on two experimentally verified reference networks in *E.coli* (Gama-Castro *et al.*, 2016), with real gene-expression data. Of particular interest in *E.coli* are the cyclic gene regulatory small sub-networks known as the 'SOS DNA repair network' (Liu *et al.*, 2016; Shen-Orr *et al.*, 2002) and the 'acid resistance regulatory network' (Foster, 2004; Shimizu, 2013).

The dataset of expression profiles used to model and evaluate the probabilistic framework consists of uniformly normalized gene-expression data in the M^{3D} database (Faith *et al.*, 2007). This compendium of data facilitates large-scale computational analyses by providing a bulk download of human-curated, computable experimental metadata, and computer-validated data for integrity. The compendium data used on *E.coli* (version 4 build 6) contain 466 microarray expression profiles of 4297 genes collected under a wide range of experimental conditions, including wild-type, pH changes, varying oxygen concentrations, antibiotics, heat shock, growth phases, different media, environmental perturbations, gene knockout (KO)/knockdown and time series at steady-state level. In addition, the expert knowledge in RegulonDB (Gama-Castro *et al.*, 2016) enabled us to assess and validate the network models.

3.1 Model organisms

3.1.1 *Escherichia coli*: SOS response model

Figure 3 shows an SOS response model that includes 9 genes with 24 edges. The pathway regulates and coordinates cell survival and repair after extensive DNA damage, which involves *lexA* and *recA* genes as principal mediators. LexA is a repressor protein dimer for the majority of DNA repair genes while the cell is healthy. On the other hand, RecA protein acts as a sensor of DNA damage that induces the response by inhibiting LexA (Liu *et al.*, 2016).

3.1.2 *Escherichia coli*: acid-resistance model

Escherichia coli has a potent acid-resistant (AR) system enabling it to survive extreme acidic environments (pH < 2.5). As such, from both medical and fermentation points of view, the physiological and molecular response to acid shock has been the subject of intense study. In this work, we consider an efficient glutamate-dependent AR regulatory network shown in Figure 4 and also reported in Foster (2004) and Shimizu (2013). The pathway has LuxR-family member GadE as the central activator. Moreover, it has at least 11 known regulatory proteins that affect the induction of the response (see Table 1) (Foster, 2004).

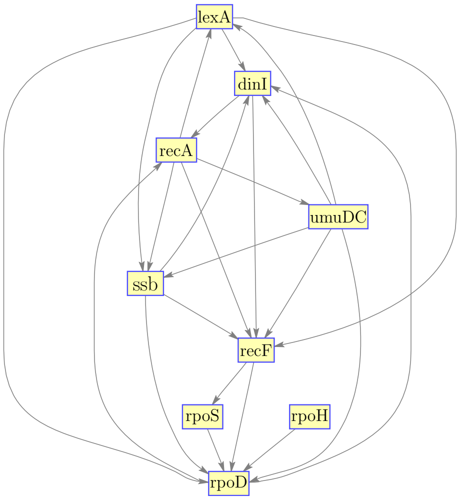

Fig. 3. Graphical representation of *E.coli* SOS DNA repair true pathway (Liu *et al.*, 2016)

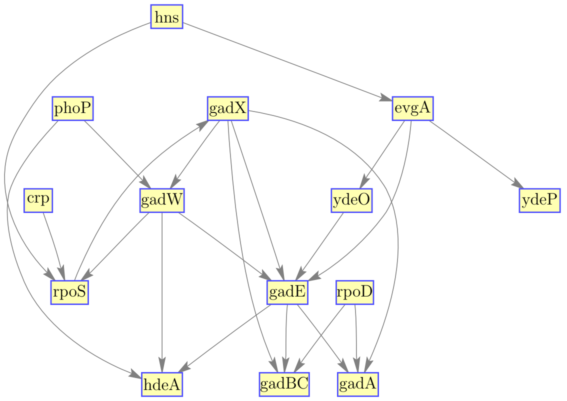

Fig. 4. Graphical representation of AR GRN (Shimizu, 2013). Overexpressed EvgA response regulator indirectly activates acid resistance through transcription of *ydeO* (i.e. EvgA → YdeO → GadE → AR). However, without overexpression (under normal inducing conditions), EvgA can directly activate *gadE* transcription

Table 1. Two-logical states distribution of SOS response network predicted marginal posteriors versus observed experimental states

Gene	Predicted marginals		Observed states	
	0	1	0	1
lexA	0.9213	0.0787	0.7876	0.2124
dinI	0.2643	0.7357	0.2918	0.7082
umuDC	0.7019	0.2981	0.7232	0.2768
recA	0.5695	0.4305	0.5687	0.4313
ssb	0.9892	0.0108	0.9292	0.0708
recF	0.1005	0.8995	0.2275	0.7725
rpoS	0.7722	0.2278	0.7296	0.2704
rpoH	0.7342	0.2658	0.7232	0.2768
rpoD	0.8670	0.1330	0.8648	0.1352

Note: At the 95% confidence interval, *t*-statistics mean difference between model-predicted marginals and observed states is expected to lie between −0.0664 and 0.0829 (i.e. for each discretization state).

3.2 Discretization of data

In probabilistic graphical models, discretizing a real-value sensor or node measurement is an integral part of the model, in order to fully account for dependencies between regulation function priors and the discretization scheme. Generally, this step is carried out if prior knowledge suggests that the underlying variables are indeed discrete, or for computational efficiency. In addition, it helps improve the robustness of data and reduce noise in the continuous variables, since discretized data can be more stable with respect to random variations of mRNA measurements (Gallo *et al.*, 2016). However, in the biological field, discretization may result in loss of information.

Finding the optimal discretization is an NP-complete problem (Chlebus and Nguyen, 1998). In the literature, several methods to discretize microarray data have been proposed (Friedman *et al.*, 2000; Gallo *et al.*, 2016; Gat-Viks *et al.*, 2006). According to Gat-Viks *et al.* (2006), for real biological gene-expression data, the variable-specific discretization scheme outperforms the global optimized single common discretization method and is generally more accurate and flexible than the standard preprocessing approaches used by Friedman *et al.* (2000). However, this flexibility can lead to over-fitting and may decrease learnability. In this work, we applied gene-specific discretization and employed mixtures of Gaussian distribution to model the relations between continuous observations on a gene variable and its discrete logical state, i.e. each Gaussian component corresponds to a specific state. Given the experimental evidence and a logical function prior for each variable, we optimize the discretization functions using the expectation–maximization (EM) learning algorithm. In each EM iteration, we infer the posterior distributions and use these to re-estimate the mixture proportions by computing the Gaussian sufficient statistics. The new discretization distributions are employed in the next iteration, and the algorithm is allowed to run iteratively until convergence.

4 Results and analysis

4.1 GRNs and data

We constructed a probabilistic FGN model representing the regulatory relations for each of the considered networks and applied the LBP inference algorithm to estimate the marginal posterior distributions on all gene logical variables. To test the prediction accuracy of our model, we computed the statistical Pearson correlation, ρ, between the inferred marginals of each gene variable G_i and the probability of G_i's observed states given by experimental observations. Furthermore, to verify our results, we compared the performance of our probabilistic model on the true networks against random networks. We implemented two network perturbation methods to obtain random networks. The first perturbation method produces a network with identical topology to the original regulatory model; however, the expression data are perturbed by uniformly redistributing the whole expression profiles of genes using the Fisher–Yates shuffle algorithm (Knuth, 2014). Entire gene profiles are swapped in order to keep each profile internally consistent, in order to be able to quantify how consistent a random set of interactions is with respect to the gene-expression compendium data. We call this method, *randomized node* perturbation because the process depicts permutation of nodes on a graph.

The second method, referred to as *randomized edge* perturbation, perturbs the regulatory network topology while preserving the node degree distribution in all nodes as well as preserving the expression profiles of individual genes. To achieve this, we implemented a simple numerical algorithm, as proposed in the work of Maslov and Sneppen (2002). The algorithm works by selecting two existing edges, (i, j) and (k, l), and reconfiguring their endpoints such that the new edges become (i, l) and (k, j). If any of these new edges already exists, then the procedure is terminated and a different pair is selected instead. This process is repeated $10 \cdot |E|$ times, where $|E|$ is the cardinality of edges in the network. The resultant network is a randomized version of the original network. In both methods, we preserve the degree of all variable nodes in a given network to rule out the impact of node degree distribution on consistency; the degree distribution is an important characteristic for nodes in biological regulatory networks (Maslov, 2008). Then, we performed multiple sampling of the resultant random networks and computed their average correlation.

In the analysis of our model, we used two- and three-quantization levels with values 0,1 and 0,1,2 states, respectively. In these contexts, we note that each state (i.e. Gaussian component) may correspond to a different range of gene-expression levels for different genes, defined by the estimated parameters of the Gaussian mixture model (GMM). Therefore, given an observation sample point for a particular gene, we say that the sample point most likely belongs to a given state with a certain probability. For instance, the *lexA* gene in the SOS response network has a normalized gene-expression data range between 7.60 and 12.70. According to the 2-states discretization, our *lexA* GMM parameters were defined by a 0-state mixture proportion of 0.7939, mean of 9.341, and variance of 0.4389, and, similarly, a 1-state mixture proportion of 0.2061, mean of 11.51, and variance of 0.1016.

4.1.1 SOS response model

Tables 1 and 2 summarize the model predictions (i.e. steady-state marginals) for the nine genes after convergence against the logical proportions of the discretized compendium data. The message updates converged at an average of 19.5 and 11.5 iterations for 2- and 3-states discretization, respectively, using two different types of initialization, i.e. random and uniform. We then constructed a 95% confidence interval using *t*-statistics to estimate the mean difference between the inferred marginals and the observed experimental states. For each level state, the expected mean difference interval contained zero. Thus, at the 95% confidence level, our results show that there is statistically no significant difference between our model-predicted marginals and the observed experimental states. Furthermore, we show the correlation plots between the variable gene G_i's inferred marginal posterior in Equation (6) at convergence and the corresponding probability of its experimental observations in Figure 5. In both plots, the correlation results are well above 0.95, thereby indicating good approximations of the observed states. We next investigated the performance of our model under the network perturbation methods. Figure 6a summarizes the comparison between inferred marginal posterior distributions and the observed experimental states for random-shuffled datasets. Figure 6b does the same for randomized topology. In general, the Pearson correlation coefficient deteriorates with increased randomization, which highlights deviation of the model predictions from the experimental observations. We deduced that the random regulatory networks are inconsistent to and do not adequately explain the DNA microarray profiles.

To further test the consistency between extant real biological data and the given GRN, we demonstrated the performance of our FGN model through a gene deletion or KO experiment. In gene KO experiments, the expression of a target protein molecule is stopped by removing the protein-coding regions from the genome. From the dataset, we identified *recA* as the most knocked-out gene in the experimental samples, having a total of 68 observations. We modified the model accordingly by fixing the state of the *recA* variable node to zero and eliminating the corresponding factor node, then, applied LBP to estimate the marginal posteriors of the other variables. Then, we compared the predictions against the observed states. Figure 7 depicts the model prediction accuracy under *recA* gene KO experimental conditions in which both 2- and 3-states discretization obtain correlation values $\rho > 0.90$. These results confirm that the given sub-network topology in *E.coli* is consistent with the gene-expression data obtained from various biological experiments.

Table 2. Three-logical states distribution of SOS response network predicted marginal posteriors versus observed experimental states

Gene	Predicted marginal			Observed states		
	0	1	2	0	1	2
lexA	0.9999	0.0	0.0001	0.7811	0.0944	0.1245
dinI	0.7112	0.0029	0.2859	0.4936	0.2553	0.2511
umuDC	0.9772	0.0	0.0228	0.7339	0.0386	0.2275
recA	0.6554	0.3429	0.0017	0.4807	0.2961	0.2232
ssb	0.0501	0.9261	0.0238	0.1567	0.7511	0.0922
recF	0.1104	0.5605	0.3291	0.1373	0.4850	0.3777
rpoS	0.8142	0.1676	0.0181	0.7704	0.1996	0.0300
rpoH	0.6626	0.3374	0.0	0.6524	0.3433	0.0043
rpoD	1.0000	0.0	0.0	0.9678	0.0193	0.0129

Note: At the 95% confidence interval, the *t*-statistics mean difference between model-predicted marginals and observed states is expected to lie between −0.0435 and 0.2229 (i.e. for 0-state level).

4.1.2 Acid-resistance model

For the AR gene network (see Fig. 4), similarly, we constructed an FGN model and applied the LBP inference algorithm to estimate the marginal beliefs. In this model, the message update algorithm converged at 14 and 20 iterations for 2- and 3-states discretization levels, respectively. Initial messages at the variable nodes were set to a uniform vector of ones. For the gene regulation functions, the Pearson correlation plots are shown in Figure 8. Here too, the correlation coefficients are >0.90. Furthermore, we tested the discrepancy of inferred marginals against randomized node- and

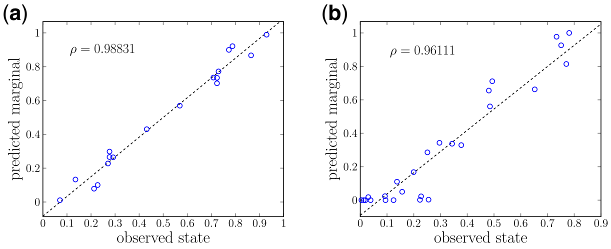

Fig. 5. Pearson correlation plots between proportions of observed states and FGN inferred marginal posteriors for SOS response network: (**a**) 2-states discretization, *P*-value = 1.6858×10^{-14} and (**b**) 3-states discretization, *P*-value = 1.7540×10^{-15}. Correlation coefficient ρ is given in each plot

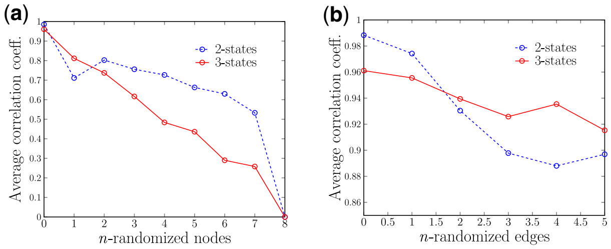

Fig. 6. Average Pearson correlation coefficient plots of *n*-randomized nodes or edges for SOS response network. Model predictions are repeatedly evaluated, and averages of correlation coefficients computed in $10 \cdot |E|$ random attempts: (**a**) randomly shuffled datasets where states are swapped between different variable nodes and (**b**) randomly shuffled *n*-edges over network structure

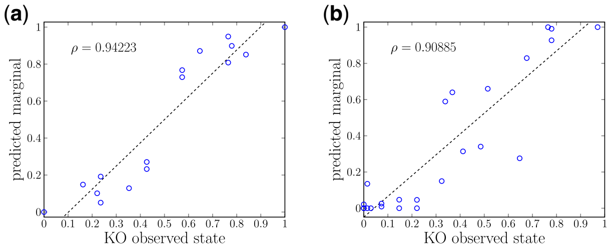

Fig. 7. Pearson correlation plots between proportions of gene experimental observations and FGN inferred beliefs in *recA* gene KO model: (**a**) 2-states discretization and (**b**) 3-states discretization. Correlation coefficient ρ is given in each plot

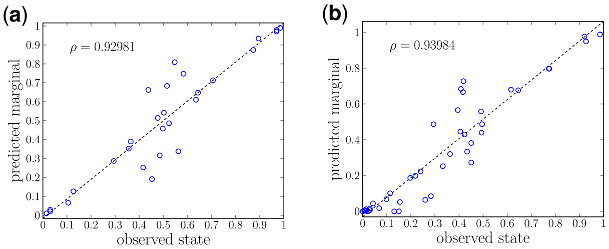

Fig. 8. Pearson correlation plots between observed states node proportions and FGN inferred marginal posteriors for AR response network: (**a**) 2-states discretization, P-value $= 8.5579 \times 10^{-13}$ and (**b**) 3-states discretization, P-value $= 2.920 \times 10^{-20}$

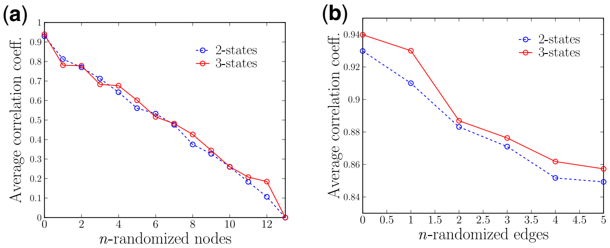

Fig. 9. Average Pearson correlation coefficient plots of n-randomized nodes or edges for AR response network. Model predictions are repeatedly evaluated, and averages of correlation coefficients are computed in $10 \cdot |E|$ random attempts: (**a**) randomly shuffled datasets where states are swapped between different variable nodes and (**b**) randomly shuffled n-edges over network structure

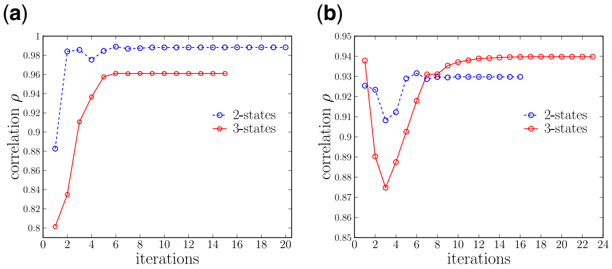

Fig. 10. Pearson correlation plots of LBP message-passing convergence with increasing iteration for both 2- and 3-states discretization levels in: (**a**) SOS response network and (**b**) AR regulatory network. CPU runtime until convergence is on average 0.26 and 0.43 s in SOS network for 2- and 3-states discretization, respectively. Accordingly, CPU runtime for AR network is on average 0.15 and 0.26 s

edge-network models, as in the case of the SOS response pathway. Similar observations were made, as depicted in Figure 9. However, in this model, we did not show performance under gene KO experimental conditions since the available expression data do not have any of the genes participating in this pathway as a deleted gene.

4.2 Software performance

4.2.1 Computational complexity

Overall, the main computational cost of running the software is the message update equation [i.e. Equations (4) and (5)], which incurs $\mathcal{O}(k^2)$ (Noorshams and Wainwright, 2013) per iteration for each pair of variable and factor nodes in the FGN model, if the state space of all g_i has k possible states. (Recall that message sent is a function of the parent. Also, we assume that the column vectors of a regulatory function $f(:, j)$ for $j = 0, 1, \ldots, k-1$, and their normalization constants have been pre-computed and stored, which can be done off-line.) This computational cost grows linearly with the number of edges. Figure 10 depicts a practical overview of the software performance and run time, where the predicted ρ is plotted against

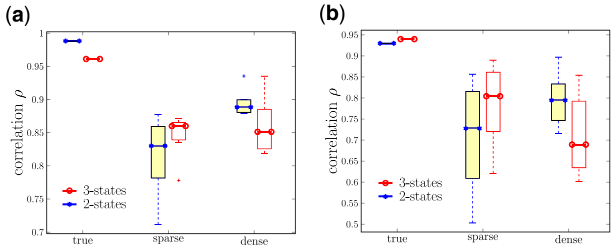

Fig. 11. Evaluation of probabilistic FGN model on GRNs with different sparsity in: (**a**) SOS response network and (**b**) AR regulatory network. Performance is compared to very sparse and dense networks for both 2-states (yellow-fill blue boxplots) and 3-states (red boxplots) discretization levels. Sparse and dense networks are obtained by deletion of edges from and addition of edges to true (original) GRNs, respectively. (Color version of this figure is available at *Bioinformatics* online.)

increasing iteration of the message-passing algorithm for both SOS response and AR regulatory networks. In Figure 10a, the messages converged after 20, and 11 iterations for 2- and 3-states discretization levels, respectively. Similarly, in Figure 10b, the messages converged after 14 and 20 iterations. An inspection of the figures reveals that the rate of convergence is at least linear (note that, linear convergence means that the error (i.e. difference between two successive ρ's) decreases exponentially), which is consistent with findings in the existing literature (Mooij and Kappen, 2007).

4.2.2 Sparse and dense networks

Here, we demonstrated the performance of our model to discriminate good GRNs from bad ones. We implemented very sparse and dense networks by deleting and adding links between gene nodes in the true GRN, in order to represent a poor network. In the SOS response network, we created sparse networks by removing 5–17 random edges. Similarly, in the AR regulatory network, we deleted between 3 and 15 random edges. On the other hand, dense networks were created by adding 5–30 random edges and 5–40 random edges, in the SOS response and AR regulatory networks, respectively. For each sparse or dense network analysis, the model predictions were repeatedly evaluated 100 times and the average Pearson correlations computed. Moreover, in the analysis of sparse gene networks, we assumed that any isolated gene node is not part of the network and set its inferred marginal to a vector of all zeros.

Figure 11 shows boxplots of the average correlation coefficients comparison between the true regulatory networks and the poor networks for both 2- and 3-states discretization levels. The SOS response network model predictions are shown in Figure 11a. In this figure, for the 2-states discretization analysis, the median correlation coefficient decreased to 0.8301 (16% decrease) and 0.8886 (10% decrease) from the true gene network correlation (i.e. 0.9883, see Fig. 5a), in sparse and dense networks, respectively. Furthermore, we observed that in the 3-states discretization analysis, the median correlation coefficient of the true network (i.e. 0.9611) decreased by 10.5% for sparse networks and 11.4% for dense networks. Similarly, Figure 11b summarizes the model predictions in the AR regulatory network. Here too, the median correlation coefficient of the sparse networks in 2-states discretization analysis was observed to be 0.7277 (21.7% decrease) and in 3-states discretization, 14.4% less than the true network correlation coefficient of 0.9398. Moreover, for the dense network analysis, the median correlation coefficients decreased by 14.5% and 25% in 2- and 3-states discretization, respectively. In general, we observed that as the number of edges added or removed was increased, the average Pearson correlation coefficients deteriorated. These results suggest that our probabilistic model can robustly separate good GRNs from poor ones.

4.2.3 Impact of discretization levels

Table 3 illustrates the performance of our probabilistic model as the number of discretization levels increases. Moreover, the corresponding average CPU runtime (this value could change depending on the processor speed of the computing machine being used.) is shown,

Table 3. FGN model predictions on increasing number of quantization levels for SOS response network and AR regulatory network, and corresponding CPU runtimes

No. of states k=	SOS network		AR network	
	Appx. correlation	CPU runtime (s)	Appx. correlation	CPU runtime
2	0.9883	0.26	0.9298	0.15 s
3	0.9611	0.43	0.9398	0.26 s
4	0.9691	8.02	0.9378	0.33 s
5	0.9405	13.08	0.8808	0.37 s
6	0.9092	19.46	0.8929	0.75 s
7	—	—	0.8999	1.53 s
8	—	—	0.8822	1.59 s
9	—	—	0.8737	3.38 s

and as expected, the network with the greater number of edges has a higher computation time. Results show a decline in predictions with a higher clustering resolution. It is noteworthy that clustering algorithms tend to be less robust with respect to the larger overlap between clusters as k increases (Rodriguez *et al.*, 2019). In addition, as we increased the number of states above 6 (i.e. in the case of the SOS response network) and above 9 (i.e. for the AR network), we observed that the discretization optimization failed due to the creation of ill-conditioned covariance matrices of the GMM. Thus, we noted, stems from the nature of the normalized gene-expression data used in this work. We observed that certain genes, especially in the SOS response network, have a data range as low as 2.8 or have multiple data points close to each other, thus leading to instances of non-invertibility of covariance matrices. Intuitively then, an appropriate value of k is limited by the properties of the dataset.

5 Conclusion

Here, we have explored a probabilistic graphical model representation of biological networks and applied a message-passing algorithm to investigate the allowable stable states and consistency between some of the existing experimentally verified pathways in the *Bacterium* genome to the diverse high-throughput real biological data. The mathematical formulation of the model describes steady-state behavior of systems where the steady-state assumption is highly adequate for a typical high-throughput experimental sampling rate. Also, in its current form, the model is already capable of handling both minimal time required for spreading perturbations in the network and undelayed feedback loops. We then applied statistical analyses to compare the variability between the model-inferred steady-state marginals and the experimental observed states. Our findings reveal a high correlation between the given network pathways and the diverse experiments gene-expression data. This implies that the small sub-networks considered are strongly supported by the measured gene-expression data. A key experimental observation is that the genetic graph has sparsely distributed and possibly long edges (Milenkovic and Vasic, 2004). Therefore, even for a genome-wide GRN, the mathematical formulations described still apply (Kschischang *et al.*, 2001).Thus, it would be interesting to see whether our framework can be employed to perceive global patterns of complex biological networks. Such patterns, however not visible on a local level, can enable us to build qualitatively new kinds of hypotheses.

One major simplification that we applied in our network model is the assumption of static GRNs, which do not adequately explain transcriptional gene expression, at least not on a cellular system-wide level. In order to enable true inference of cellular functions and organization, future methods on more complex models that consider the fluidity of biological networks (i.e. changing networks with time, context and conditions), temporality and multi-omics data would be required. Furthermore, in this work, we applied the modeling of distributions over discrete functions, primarily since most of the current biological knowledge on regulatory relations and transcriptional switches is essentially qualitative. We note that in its current form, our model cannot predict actual non-discretized gene-expression levels. However, in order to do so, the framework can readily be adapted by defining continuous probability distribution over the regulatory function, f_i. This would require much more data to adequately learn the regulatory relations toward more significant results (Gallo *et al.*, 2016).

Funding

This work was supported by the National Science Foundation under Award No. [OIA-1656006]; and matching support from the State of Kansas through the Kansas Board of Regents.

Conflict of Interest: none declared.

References

Aji,S.M. and McEliece,R.J. (2000) The generalized distributive law. *IEEE Trans. Inf. Theory*, **46**, 325–343.

Chlebus,B.S. and Nguyen,S.H. (1998) On finding optimal discretizations for two attributes. In: International Conference on Rough Sets and Current Trends in Computing. pp. 537–544. Springer, Berlin Heidelberg.

Chouvardas,P. *et al.* (2016) Inferring active regulatory networks from gene expression data using a combination of prior knowledge and enrichment analysis. *BMC Bioinformatics*, **17**, 181.

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

Covert,M.W. *et al.* (2004) Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, **429**, 92–96.

de Hoon,M. *et al.* (2002) Inferring gene regulatory networks from time-ordered gene expression data using differential equations. In: International Conference on Discovery Science. pp. 267–274. Springer, Berlin Heidelberg.

Faith,J.J. *et al.* (2007) Many microbe microarrays database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.*, **36**, D866–D870.

Foster,J.W. (2004) *Escherichia coli* acid resistance: tales of an amateur acidophile. *Nat. Rev. Microbiol.*, **2**, 898–907.

Fratello,M. *et al.* (2015) A multi-view genomic data simulator. *BMC Bioinformatics*, **16**, 151.

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Gallo,C.A. *et al.* (2016) Discretization of gene expression data revised. *Brief. Bioinform.*, **17**, 758–770.

Gama-Castro,S. *et al.* (2016) RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.*, **44**, D133–D143.

Gat-Viks,I. *et al.* (2006) A probabilistic methodology for integrating knowledge and experiments on biological networks. *J. Comput. Biol.*, **13**, 165–181.

Hartemink,A.J. *et al.* (2001) Combining location and expression data for principled discovery of genetic regulatory network models. *Biocomputing*, **2002**, 437–449.

Hoops,S. *et al.* (2006) COPASI–a complex pathway simulator. *Bioinformatics*, **22**, 3067–3074.

Hughes,T.R. *et al.* (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Karlebach,G. and Shamir,R. (2008) Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.*, **9**, 770–780.

Klipp,E. *et al.* (2008) *Systems Biology in Practice: Concepts, Implementation and Application*. John Wiley & Sons, New York.

Knuth,D.E. (2014). *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional, Boston.

Kschischang,F.R. *et al.* (2001) Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory*, **47**, 498–519.

Liu,F. *et al.* (2016) Inference of gene regulatory network based on local Bayesian networks. *PLoS Comput. Biol.*, **12**, e1005024.

Lovrics,A. *et al.* (2014) Boolean modelling reveals new regulatory connections between transcription factors orchestrating the development of the ventral spinal cord. *PLoS One*, **9**, e111430.

Madhamshettiwar,P.B. *et al.* (2012) Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome Med.*, **4**, 41.

Marbach,D. *et al.*; The DREAM5 Consortium. (2012) Wisdom of crowds for robust gene network inference. *Nat. Methods*, **9**, 796–804.

Maslov,S. (2008) Topological and dynamical properties of protein interaction networks. In: Panchenko,A. and Przytycka,T. (eds.) *Protein-Protein Interactions and Networks*. Springer, London, pp. 115–137.

Maslov,S. and Sneppen,K. (2002) Specificity and stability in topology of protein networks. *Science*, **296**, 910–913.

Milenkovic,O. and Vasic,B. (2004) Information theory and coding problems in genetics. In: *Information Theory Workshop*. pp. 60–65. IEEE, San Antonio, Texas.

Mooij,J.M, and Kappen,H.J. (2007) Sufficient conditions for convergence of the sum–product algorithm. *IEEE Trans. Inf. Theory*, **53**, 4422–4437.

Murphy,K.P. *et al.* (1999) Loopy belief propagation for approximate inference: an empirical study. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. pp. 467–475. Morgan Kaufmann Publishers Inc, Stockholm, Sweden.

Noorshams,N. and Wainwright,M.J. (2013) Stochastic belief propagation: a low-complexity alternative to the sum-product algorithm. *IEEE Trans. Inf. Theory*, **59**, 1981–2000.

Pearl,J. (2014) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, Amsterdam.

Prill,R.J. *et al.* (2010) Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PLoS One*, **5**, e9202.

Rodriguez,M.Z. *et al.* (2019) Clustering algorithms: a comparative approach. *PLoS One*, **14**, e0210236.

Shen-Orr,S.S. *et al.* (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, **31**, 64–68.

Shimizu,K. (2013) Metabolic regulation of a bacterial cell system with emphasis on *Escherichia coli* metabolism. *ISRN Biochem.*, **2013**, 1–47.

Stolovitzky,G. *et al.* (2007) Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Ann. N. Y. Acad. Sci.*, **1115**, 1–22.

Tripathi,S. *et al.* (2017) sgnesR: an R package for simulating gene expression data from an underlying real gene network structure considering delay parameters. *BMC Bioinformatics*, **18**, 325.

Yedidia,J.S. *et al.* (2005) Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theory*, **51**, 2282–2312.

Yu,X. *et al.* (2014) A computational method of predicting regulatory interactions in *Arabidopsis* based on gene expression data and sequence information. *Comput. Biol. Chem.*, **51**, 36–41.