INFORMATIVE STRUCTURE PRIORS: JOINT LEARNING OF DYNAMIC REGULATORY NETWORKS FROM MULTIPLE TYPES OF DATA

ALLISTER BERNARD, ALEXANDER J. HARTEMINK

Duke University, Dept. of Computer Science, Box 90129, Durham, NC 27708

We present a method for jointly learning dynamic models of transcriptional regulatory networks from gene expression data and transcription factor binding location data. Models are automatically learned using dynamic Bayesian network inference algorithms; joint learning is accomplished by incorporating evidence from gene expression data through the likelihood, and from transcription factor binding location data through the prior. We propose a new informative structure prior with two advantages. First, the prior incorporates evidence from location data probabilistically, allowing it to be weighed against evidence from expression data. Second, the prior takes on a factorable form that is computationally efficient when learning dynamic regulatory networks. Results obtained from both simulated and experimental data from the yeast cell cycle demonstrate that this joint learning algorithm can recover dynamic regulatory networks from multiple types of data that are more accurate than those recovered from each type of data in isolation.

1 Introduction and motivation

Discovering networks of transcriptional regulation is an important problem in molecular biology, and progress toward this goal has been accelerated by the advent of new technologies for collecting high-throughput data. Data collected using different technologies offer different perspectives on a problem, but jointly analyzing such data in a single framework enables a consensus perspective to emerge. In addition, joint analysis is likely to produce more accurate results since noise characteristics and biases of the various technologies should be largely independent. Here, we present a framework for jointly learning dynamic models of transcriptional regulatory networks from both gene expression data and transcription factor binding location data.

Most early research on automatic learning of transcriptional regulatory networks used only gene expression data.[1,2] However, recent simulation studies suggest that regulatory networks learned from gene expression data alone can be considerably obscured by the recovery of spurious interactions when the number of observations is small,[3,4] although a few methods have been developed to address this problem.[4] Joint learning from multiple types of data can alleviate the problem further.

Joint learning in the context of transcriptional regulation has primarily developed around two somewhat related approaches. In one approach, various types of data are used to identify *sets of genes* that interact together in the cell (pathways), share common roles (processes), or are regulated in concert (modules). Recent noteworthy examples include Segal et al.[5] and Bar-Joseph et al.[6] In the other approach, various types of data are used to supplement gene expression data when learning regulatory network models directly. This latter approach was developed by Hartemink et al. and applied in the context of supplemental data describing transcription factor binding location.[7] More recently, Imoto et al. proposed a similar method that applies

in the context of supplemental data describing interactions that are restricted to be binary and symmetric,[8] such as the presence of a protein-protein interaction.[9]

Hartemink et al.[7] had two significant limitations that we overcome here. First, because Bayesian networks must be directed acyclic graphs, the resultant models were previously incapable of representing feedback and other dynamic processes. In contrast, here we use a dynamic Bayesian network (DBN) framework to learn regulatory network models from time series data. DBNs are a class of Bayesian network models that permit cyclic structures like regulatory feedback loops; DBNs have been advocated for use in a biological context,[10] and have been used before to analyze time series data in contexts from transcription in *E. coli*[11] to brain electrophysiology in songbirds.[12]

Second, because computational efficiency is a serious concern, the location data were previously treated as infallible: models that failed to include an edge where the location data suggested one should be were eliminated from consideration *a priori*. In formal terms, the prior probability for such networks was forced to zero. While this enabled computationally efficient search, it is inconsistent with the notion that genomic data are generally quite noisy. In contrast, here we present a new informative prior over network structures that is capable of incorporating the location data using a smooth probabilistic model: location data provides evidence as to whether a regulatory relationship exists, and now the more significant the location data (lower the p-value), the more likely the edge is to be included. As a consequence, this prior is subtler and more robust; nevertheless, it is designed to be factorable in the context of a DBN, enabling computationally efficient local search. In fact, we can learn a DBN model using the new structure prior in the same amount of time as before. Moreover, unlike in Imoto et al.,[8] our prior is suited to handle the kinds of asymmetric interactions that exist between regulators and their targets.

2 Modeling framework

A Bayesian network encodes a probability distribution over a set of random variables $\boldsymbol{X} = \{X_1, \ldots, X_N\}$. The encoding of this probability distribution consists of two components: a network structure S and a set of parameters Θ. The network structure S describes the qualitative nature of the dependencies between the random variables in the form of a directed acyclic graph; the vertices of the graph represent the random variables in $\boldsymbol{X}$ and the directed edges represent the dependencies between those variables. In particular, each variable X_i is assumed to be independent of its non-descendants given its set of parents, denoted $\mathbf{Pa}(X_i)$. Under such a Markov assumption, the joint probability distribution can be written:

$$\mathcal{P}(\boldsymbol{x}) = \prod_{i=1}^{N} \mathcal{P}(x_i \mid \mathbf{pa}(X_i)) \tag{1}$$

where lowercase variables denote values of the corresponding uppercase random variables. The set of parameters Θ describes the quantitative nature of the dependencies between the random variables by characterizing the individual probability distributions in this product.

2.1 Dynamic Bayesian networks

A dynamic Bayesian network[13] extends the notion of a Bayesian network to model the stochastic evolution of a set of random variables over time; the structure of a DBN thus describes the qualitative nature of the dependencies that exist between variables in a temporal process. We use $X_i[t]$ to denote the random variable X_i at time t and the set $\boldsymbol{X}[t]$ is defined analogously. The evolution of the temporal process is assumed to occur over discrete time points indexed by the variable $t \in \{1, \ldots, T\}$. Under such an assumption, we now have $T \times N$ interacting random variables where previously we had N. To simplify the situation, we make two further assumptions. First, we assume that each variable can only depend on variables that temporally precede it. This fairly innocuous assumption allows us to model natural phenomena like feedback loops, but still guarantee that the underlying graph will be acyclic. Second, we assume that the process is a stationary first-order Markov process, which means that $\mathcal{P}(\boldsymbol{X}[t] \mid \boldsymbol{X}[t-1], \ldots, \boldsymbol{X}[1], t) = \mathcal{P}(\boldsymbol{X}[t] \mid \boldsymbol{X}[t-1])$. This assumption is somewhat less innocuous, and we discuss relaxations later. Given these two assumptions, the resultant joint probability distribution can be written:

$$\mathcal{P}(\boldsymbol{x}[1], \ldots, \boldsymbol{x}[T]) = \prod_{i=1}^{N} \left[\mathcal{P}(x_i[1]) \prod_{t=2}^{T} \mathcal{P}(x_i[t] \mid \mathbf{pa}(X_i[t])) \right] \tag{2}$$

where we note that the first-order Markov assumption means the variables in the set $\mathbf{Pa}(X_i[t])$ are a subset of $\boldsymbol{X}[t-1]$. The underlying acyclic graph with $T \times N$ vertices can now be compactly represented by a graph with N vertices that is permitted to have cycles (see Figure 1 for an example).

2.2 Learning the structure of a dynamic Bayesian network

The goal when learning the structure of a dynamic Bayesian network is to identify the network structure S that is most probable given some observed data D; in the context of a DBN, the data D typically consists of the T observations of the N variables. The notion of the most probable network structure is made formal by the Bayesian scoring metric (BSM), which is simply the log posterior probability of S given D:[14]

$$\mathsf{BSM}(S : D) = \log \mathcal{P}(S|D) = \log \mathcal{P}(D|S) + \log \mathcal{P}(S) + c \tag{3}$$

where the constant c is the same for all structures and can be safely ignored. In a fully Bayesian treatment, the calculation of the log likelihood, $\log \mathcal{P}(D|S)$, involves marginalizing over the distribution of possible parameters Θ, which is analytically

tractable when the variables in the network are discrete,[14] as we assume here. Because the expression in (2) is factorable as a product over the variables, the resultant closed-form expression for the log marginal likelihood can be written as a sum of terms where each term corresponds to one variable. Thus, a local change to the network structure—adding or deleting a single edge—affects only one term in this sum.

An especially common choice for the log prior over structures, $\log \mathcal{P}(S)$, is to assume that it is uninformative: every structure is equally likely; in this case, the prior term can be safely ignored since it is the same for all structures. In the rare instance where an informative prior is chosen, it is typically hand-constructed by domain experts.[14] In the next section, we develop a new approach for constructing an informative prior over regulatory network structures automatically from location data.

3 Informative structure priors

Transcription factor binding location data provides (noisy) evidence as to the existence of a regulatory relationship between a transcription factor and genes in the genome. This evidence is reported as a p-value, and the probability of an edge being present in the true regulatory network is inversely related to this p-value: the smaller the p-value, the more likely the edge is to exist in the true structure. A more precise formulation of this relationship is provided below.

3.1 Probability of an edge being present

We first need to derive a function to map p-values to corresponding probabilities of edges being present in structure S. Let us define the p-value for the location data corresponding to edge E_i in terms of a random variable P_i defined on the interval $[0, 1]$. In this interval, P_i has been previously assumed to be exponentially distributed[16] if the edge E_i is present in S, and uniformly distributed if the edge E_i is absent from S (by the definition of a p-value). Formally, we have $\mathcal{P}_\lambda(P_i = p \mid E_i \in S) = \lambda e^{-\lambda p}/(1 - e^{-\lambda})$, where λ is the parameter controlling the scale of the truncated exponential distribution, and $\mathcal{P}(P_i = p \mid E_i \notin S) = 1$.

Let us use β to denote $\mathcal{P}(E_i \in S)$, the probability that edge E_i is present before observing the corresponding p-value. Using Bayes rule, we can show that the probability that edge E_i is present after observing the corresponding p-value is:

$$\mathcal{P}_\lambda(E_i \in S \mid P_i = p) = \frac{\lambda e^{-\lambda p}\beta}{\lambda e^{-\lambda p}\beta + (1 - e^{-\lambda})(1 - \beta)} \tag{4}$$

As the parameter λ increases, the mass of this distribution becomes more concentrated at smaller values of P_i; conversely, as λ decreases, the distribution spreads out and flattens. The role of the parameter λ can be more clearly understood by considering the value p^* obtained by solving the equation $\mathcal{P}_\lambda(E_i \in S \mid P_i = p^*) =$

$\mathcal{P}_\lambda(E_i \notin S \mid P_i = p^*)$, which yields:

$$p^* = \frac{-1}{\lambda} \log \left[\frac{(1 - e^{-\lambda})(1 - \beta)}{\lambda \beta} \right] \tag{5}$$

For any fixed value of λ, an edge E_i is more likely to be present than absent if the corresponding p-value is below this critical value p^* (and vice versa). As we increase the value of λ, the value of p^* decreases and we become more stringent about how low a p-value must be before we consider it as prior evidence for edge presence. Conversely, as λ decreases, p^* increases and we become less stringent; indeed, in the limit as $\lambda \to 0$, we can show that $\mathcal{P}_\lambda(E_i \in S \mid P_i = p) \to \beta$ independent of p, revealing that if we have no confidence in the location data, the probability that edge E_i is present is the same value β both before and after seeing the corresponding p-value, as expected. Thus, λ acts as a tunable parameter indicating the degree of confidence in the evidence provided by the location data; this allows us to model our belief about the noise level inherent in the location data and correspondingly, the amount of weight its evidence should be given.

3.2 Bayesian marginalization over parameter λ

One approach to suitably weighing the evidence of the location data would be to somehow select a single value for λ, either by guessing or by some other heuristic like finding the value of λ that corresponds to a certain "magic" value for p^* like 0.001. Instead, we adopt a more robust Bayesian approach that avoids the selection of a single value and instead marginalizes over λ. For convenience, we assume that λ is uniformly distributed over the interval $[\lambda_L, \lambda_H]$ and then integrate λ out of (4) to yield:

$$\mathcal{P}(E_i \in S \mid P_i = p) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_L}^{\lambda_H} \frac{\lambda e^{-\lambda p} \beta}{\lambda e^{-\lambda p} \beta + (1 - e^{-\lambda})(1 - \beta)} \, \mathrm{d}\lambda \tag{6}$$

Although (6) cannot be solved analytically, it can be solved numerically for fixed $P_i = p$. Since we have a finite set of p-values for a given set of location data, we precompute this integral for each p-value and store the results in a table for later use. The computational overhead associated with marginalizing over λ is thus constant. The net effect of marginalization is an edge probability distribution that is a smoother function of the reported p-values than without marginalization (the tail is much heavier; for a visual depiction, please see the figure in the supplemental material).

3.3 Prior probability of a structure

We express the complete log prior probability over structures using the following edge-wise decomposition:

$$\log \mathcal{P}(S) = \sum_{E_j \in S} \log \mathcal{P}(E_j \in S \mid P_j = p) + \sum_{E_k \notin S} \log \mathcal{P}(E_k \notin S \mid P_k = p)) \tag{7}$$

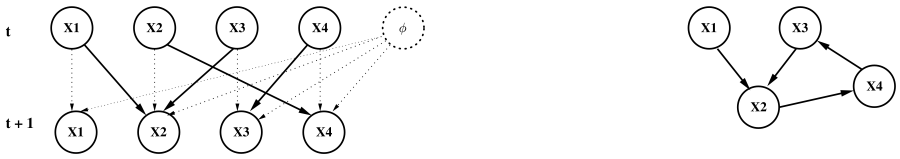

Figure 1. Simplified schematic of a first-order Markov DBN model of the cell cycle. On the left, variables $X1$ through $X4$ are shown both at time t and $t+1$; variable ϕ represents the cell cycle phase; dashed edges are stipulated to be present whereas solid edges are recovered by the learning algorithm. On the right, a compact representation of the same DBN model in which the cycle between $X4$, $X3$, and $X2$ is apparent.

where the term corresponding to the normalizing constant has been dropped since it is the same for all structures. Analogous to the likelihood calculations, the calculations of the prior under this formulation are computationally efficient: a local change to the network structure affects only one term in this sum. As a result, we need not recompute the entire prior with each local change.

Note that in the absence of location data pertaining to a particular edge, we simply use the probability $\mathcal{P}(E_i \in S) \equiv \beta$ for that edge. Our informative prior is thus a natural generalization of traditional priors: in the absence of any location data whatsoever, the prior probability of a network structure is exponential in the number of edges in the graph, with edges favored if we choose $\beta > 0.5$ and edges penalized if we choose $\beta < 0.5$. In the special case where $\beta = 0.5$, the prior over structures is uniform.

4 Learning dynamic models of the cell cycle

We assume above that the stochastic process regulating the expression of genes throughout the cell cycle is stationary. This poses a bit of a problem since we may have a different underlying genetic regulatory network during each phase of the cell cycle. To overcome this problem, we employ an additional variable ϕ that can be used by the model to explain how each variable's regulators depend on the cell cycle phase. The phase variable ϕ is multinomial and has as many states as there are phases in the cell cycle, allowing us to model a different stationary process within each phase. If we can label each of the time points with the appropriate phase, the inference problem reduces to learning network structure with complete data. We prefer this option to the alternative of learning a hidden phase variable because in our context, the quantity of expression data that is available is quite limited; besides, the state of ϕ changes smoothly and predictably so labeling each time point with the appropriate phase is fairly straightforward. A simplified schematic of such a DBN model of the cell cycle is depicted in Figure 1.

Because space is quite limited here, we provide only a brief description of the basic structure of each of our experiments (for further details, please see the supplemental material). The experimental gene expression data are discretized into three

states using interval discretization;[4] the simulated data are discretized into two states because the generating model is Boolean. The discretized data in each case are used to compute the log likelihood component of the BSM. The log prior component is computed from the location data p-values using (6) and (7). The parameter λ is marginalized over a wide range of values, setting $\lambda_L = 1$ to avoid problems near zero ($\lambda_L = 1$ corresponds to $p^* = 0.459$) and $\lambda_H = 10000$ to avoid problems near infinity ($\lambda_H = 10000$ corresponds to $p^* = 0.001$). We set $\beta = 0.5$ so that edges for which we have no location data are equally likely to be present or absent in the graph; as a consequence, without location data, edge presence in the graph depends on expression data alone. We use simulated annealing as a heuristic search method to identify network structures with high scores because learning optimal Bayesian networks is known to be NP-hard.[14] The output of our DBN inference algorithm is the network structure with the highest BSM score among all those visited by the heuristic search during its execution.

4.1 Results using simulated data

We use simulated data from a synthetic cell cycle model to evaluate the accuracy of our algorithm and determine the relative utility of different quantities of available gene expression data. The synthetic cell cycle model involves 100 genes and a completely different regulatory network operates in each of the three phases of the cycle. The 100 genes include synthetic transcription factors, only some of which are involved in the cell cycle, and only some of which have simulated location data available. The target genes of the transcription factors are sometimes activated and sometimes repressed; some are under cell cycle control, but many are not. In addition, we include a number of additional genes whose expression is random and not regulated by genes in the model. The simulated gene expression data is generated using the (stochastic) Boolean Glass gene model.[17] Noisy p-values for the simulated location data associated with a subset of the regulators are generated with noise models of varying intensity.

We repeatedly conduct the following three experiments: score network structures with expression data alone, ignoring the log prior component $\log \mathcal{P}(S)$ in (3); score network structures with location data alone, using the prior component of the score as given by (7) and ignoring the log likelihood component $\log \mathcal{P}(D|S)$ in (3); and score network structures with both expression and location data. We use these experiments to evaluate the effects of location data with different noise characteristics, expression data of varying quantity, and different choices for β.

Each of our experiments is conducted on five independently-generated synthetic data sets and results are averaged over those five data sets. Most of the results are presented in the supplemental material, but Figure 2 offers a representative result. The vertical axis measures the (average) total number of errors: the sum of false positives and false negatives in the learned network; the total number of errors relative to the synthetic network in our experiment can range from 0 to 10000. As expected,

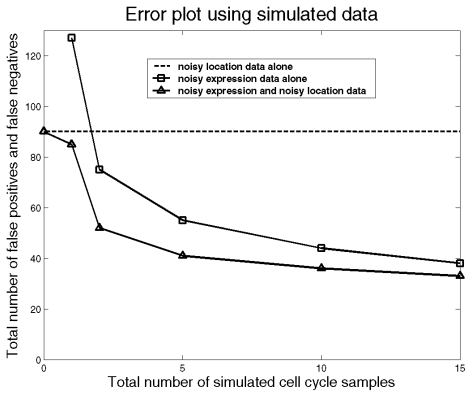

Figure 2. Total number of errors while learning a synthetic cell cycle network using (noisy simulated) expression and location data, separately and with both types of data together. The graph shows the effect of increasing the number of cell cycles worth of expression data, both with and without location data. The dashed horizontal line represents learning using location data alone.

the total number of errors drops sharply as the amount of available expression data increases. The figure demonstrates that our joint learning algorithm consistently reduces the total number of false positives and false negatives learned when compared to the error rate obtained using either expression or location data alone. Also, observe that the availability of location data means that we require typically only half as much expression data to achieve the same error rate as would be achieved with expression data alone, suggesting that the availability of location data can be used to compensate for small quantities of expression data.

4.2 Results using experimental data

We next apply our joint learning algorithm to uncover networks describing the regulation of transcription during the cell cycle in yeast. We use publicly available cell cycle gene expression data[18] and transcription factor binding location data.[19] The gene expression data consist of 69 time points collected over 8 cell cycles. Since these belong to different phases, the resultant number of time points in each phase is quite small. As a consequence, we choose to use only three states for the phase variable, by splitting the shortest phase G_2 in half and lumping the halves with the adjacent phases. Thus, the three states of our phase variable correspond roughly to G_1, $S + G_2$, and $G_2 + M$. To generate a phase label for each time point, we select characteristic genes known to be regulated during specific phases.[18] Guided by the expression of these characteristic genes, we can assign a phase label to each time point. This is done separately for each of the four synchronization protocols in the dataset (alpha, cdc15, cdc28, and elu).

We select a set of 25 genes, of which 10 are known transcription factors for

Table 1. Comparison of the highest scoring networks found in four different experiments with the gold standard network. As discussed in the text, the gold standard contains edges from only the 10 variables for which both location and expression data is available.

Experiment	**TP**	**TN**	**FP**	**FN**
Expression data only	7	181	20	32
Location data only	25	184	17	14
Expression and location data (old prior)	23	187	19	11
Expression and location data (new prior)	28	189	12	11

which we have available location data. The only important cell cycle transcription factor with location data missing from this set is FKH2; we are not able to use it in our analysis because expression data is missing for many of the time points. The remaining 15 genes in our set are selected on the basis of their known regulation by one or more of these 10 transcription factors. We apply our DBN inference algorithm on this set of 25 variables. Just as with the simulated data, we learn network structures using expression data alone, using location data alone, and jointly from both expression and location data. In the latter case, we evaluate both our old prior,[7] as before with a hard cutoff of $p = 0.001$, and our new informative prior.

As an evaluation criterion (which is more difficult in this context than in the synthetic network context), we create a "gold standard" network consisting of the set of edges that are known to exist from one of the 10 transcription factors with both expression and location data to any one of the other 24 genes in our set; we do not count edges from the other 15 genes when comparing with our gold standard since it would be difficult to determine whether recovered edges are true or false positives, and whether omitted edges are true or false negatives. The gold standard comes from a compiled list of evidence in the literature and from the Saccharomyces Genome Database (*http://www.yeastgenome.org*), but we have tried to ensure that it depends on neither the specific expression data nor the specific location data used in these experiments. Note also that the gold standard is likely not the true underlying regulatory network, but rather is the best we can do given the current understanding of the yeast cell cycle (a bronze standard?).

With these caveats in place, Table 1 shows the total number of positives and negatives that are true and false for the networks found in the four experiments, with respect to the gold standard network. We see that the location data by itself does noticeably better than the expression data, suggesting that this particular set of location data is quite insightful and/or that this particular set of expression data is quite limited in its quantity and quality. Despite the relatively poor performance of the expression data when considered in isolation, when we use our new informative prior to include evidence from the expression data along with the location data, the number of false positives and the number of false negatives are both reduced; in contrast, the old prior reduces the number of false negatives and increases the number of true negatives, but also increases the number of false positives and reduces the

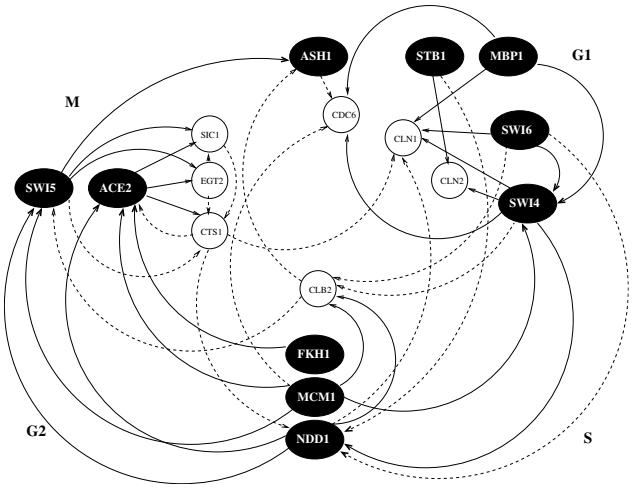

Figure 3. Partial regulatory network recovered using expression data from Spellman et al. and location data from Lee et al. Shaded elliptical nodes are transcription factors for which location data is available. Unshaded circular nodes are genes for which no location data is available. Solid edges represent interactions that have been independently verified in the literature. Dashed edges represent interactions that have not been verified; either the edge is incorrect or the evidence from the literature is inconclusive. Observe the cyclic regulation of transcription factors across phases of the cell cycle.

number of true positives. In contrast, the new prior uniformly outperforms all the other three methods.

From Table 1, we see that combining expression and location data with our new informative prior results in three fewer false negatives as compared to location data alone. These three are the binding of FKH1 to ACE2 (p-value$= 0.0058$), SWI4 to CLN2 (p-value$= 0.005$), and ACE2 to SIC1 (p-value$= 0.0095$). These edges are detected because while the evidence of the location data in isolation is just below threshold for inclusion, during the joint learning it is reinforced with evidence from expression data. Among the supposed false positives, we observe that both location and expression data provide evidence for the regulation of cyclin PCL2 by the transcription factor SWI6 although there is no known evidence of this interaction in the literature. Another interesting case is the regulation of transcription factor FKH1 by the transcription factor MBP1: although this interaction is detected by expression data alone, it is not detected when both location and expression data are used because the corresponding p-value of 0.93 is so high that the quantity of expression data is insufficient to overcome the location data evidence against inclusion of the edge.

Figure 3 shows part of the regulatory network recovered using our complete joint learning algorithm. The partial network consists of the 10 transcription factors with location and expression data, along with 7 of the other 15 genes selected at random; we do not show the full network to keep the figure as uncluttered as possible (for the full network, please see the supplemental material). The transcription

factors are arranged according to the phase of the cell cycle in which they are maximally expressed. The figure shows that the important cell cycle transcription factors regulate each other in a cyclic fashion as has previously been observed.[19] The figure also shows that our model sometimes detects interactions between genes whose expression is correlated (e.g., CTS1, EGT2, and SIC1). Although these genes are not known to exhibit direct regulatory relationships, they are related in the sense that they are co-regulated by ACE2 and SWI5.

5 Conclusion and future work

In this paper, we demonstrate the benefits of recovering dynamic models of transcriptional regulatory networks by jointly learning from both gene expression data and transcription factor binding location data. Our method uses a new, factorable informative prior over network structures to incorporate location data into the learning process. Because location data provides direct evidence regarding the presence of an edge in a regulatory network, it fits well into the framework of our informative prior, and DBNs more generally. With this joint learning framework in place, we show that supplementing expression data with location data is useful both in increasing the accuracy of recovered networks, and in reducing the quantity of expression data needed to achieve an accuracy comparable to that of expression data alone. Since expression data is fairly expensive to generate, it is promising that the relative utility of the data can be further enhanced by combining it with other types of data. Different sources of data will have different noise characteristics and so may be able to reduce the overall error present in the learned network structure. We expect such joint learning techniques will become increasingly relevant in computational biology, especially as data of greater quality and diversity become available.

From a computational perspective, our method should scale well to networks of hundreds of interacting variables, as we have demonstrated here and elsewhere.[20] The primary limitation is not computational but statistical, and not so much with respect to the number of variables but with respect to the number of parents for each variable. As this number increases, larger and larger quantities of data are needed to learn an accurate DBN model.[4] On a related note, while nothing precludes us computationally from modeling a higher-order Markov process, we are constrained statistically by the limited quantity of available time-series expression data.

Note that although the interactions in our graphs can be oriented unambiguously (because time cannot flow backwards), that does not necessarily imply that the interactions are causal since we cannot account for cellular interactions that have not been measured. This can lead to latent variable problems in which we may learn spurious interactions between observed variables: a set of variables may appear to be correlated simply because we cannot observe their latent common cause. One of the main hopes of this line of research is that more direct causal information from alternative assays like transcription factor binding location data and protein-protein interaction data will ameliorate this problem if we can include them in the analysis

framework in a principled way.

A number of directions remain in developing sophisticated joint learning methods for elucidating dynamic networks like the cell cycle. Most critically, we would like to incorporate a wider range of other sources of data like protein expression data, protein-protein interaction data, and DNA sequence data. Protein expression data can be added straightforwardly, but is not yet widely available. Nariai et al.[9] have developed a method for learning when to merge co-expressed regulators into complexes based on protein-protein interaction data that might be amenable to further generalization. Greater connection with the module approaches of Segal et al.[5] would also be fruitful.

Supplemental material is available from *http://www.cs.duke.edu/∼amink/*.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their suggestions, which led to a number of improvements in the manuscript. AJH would like to gratefully acknowledge the support of the National Science Foundation under its CAREER award program.

References

1. A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. In *PSB*, p. 422–433, 2001.
2. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. In *RECOMB*, ACM, 2000.
3. D. Husmeier. *Bioinformatics*, 19:2271–2282, 2003.
4. J. Yu, A. Smith, P. Wang, A. Hartemink, and E. Jarvis. *Bioinformatics*, to appear, 2004.
5. E. Segal, R. Yelensky, and D. Koller. *Bioinformatics*, 19:i273–i282, 2003.
6. Z. Bar-Joseph et al. *Nature Biotechnology*, 21(11):1337–1341, 2003.
7. A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. In *PSB*, p. 437–449, 2002.
8. S. Imoto, T. Higuchi, T. Goto, and K. Tashiro. In *CSB*, IEEE, 2003.
9. N. Nariai, S. Kim, S. Imoto, and S. Miyano. In *PSB*, p. 384–395, 2004.
10. K. Murphy and S. Mian. University of California at Berkeley, 1999.
11. I. Ong, J. Glasner, and D. Page. *Bioinformatics*, 18:S241–S248, 2002.
12. A. Smith, J. Yu, T. Smulders, A. Hartemink, and E. Jarvis. In preparation, 2004.
13. N. Friedman, K. Murphy, and S. Russell. In *Proc. 14th UAI*, p. 139–147, 1998.
14. D. Heckerman. In M. I. Jordan, editor, *Learning in Graphical Models*, p. 301–354, Kluwer Academic Publishers, 1998.
15. T. Hughes et al. *Cell*, 102:109–126, 2000.
16. E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. In *RECOMB*, ACM, 2002.
17. R. Edwards and L. Glass. *Chaos*, 10:691–704, 2000.
18. P. Spellman et al. *Mol. Biol. Cell*, 9:3273–3297, 1998.
19. T. Lee et al. *Science*, 298:799–804, 2002.
20. A. Smith, E. Jarvis, and A. Hartemink. *Bioinformatics*, 18:S216–S224, 2002.