From Promoter Sequence to Expression: A Probabilistic Framework

Eran Segal
Computer Science Department
Stanford University
Stanford, CA 94305-9010
eran@cs.stanford.edu

Yoseph Barash
School of Computer Science & Engineering
Hebrew University of Jerusalem
Jerusalem, Israel 91904
hoan@cs.huji.ac.il

Itamar Simon
Whitehead Institute
Cambridge, MA 02142
simon@wi.mit.edu

Nir Friedman*
School of Computer Science & Engineering
Hebrew University of Jerusalem
Jerusalem, Israel 91904
nir@cs.huji.ac.il

Daphne Koller
Computer Science Department
Stanford University
Stanford, CA 94305-9010
koller@cs.stanford.edu

ABSTRACT

We present a probabilistic framework that models the process by which transcriptional binding explains the mRNA expression of different genes. Our joint probabilistic model unifies the two key components of this process: the prediction of gene regulation events from sequence motifs in the gene's promoter region, and the prediction of mRNA expression from combinations of gene regulation events in different settings. Our approach has several advantages. By learning promoter sequence motifs that are directly predictive of expression data, it can improve the identification of binding site patterns. It is also able to identify combinatorial regulation via interactions of different transcription factors. Finally, the general framework allows us to integrate additional data sources, including data from the recent binding localization assays. We demonstrate our approach on the cell cycle data of Spellman *et al.*, combined with the binding localization information of Simon *et al.* We show that the learned model predicts expression from sequence, and that it identifies coherent co-regulated groups with significant transcription factor motifs. It also provides valuable biological insight into the domain via these co-regulated "modules" and the combinatorial regulation effects that govern their behavior.

1. Introduction

A central goal of molecular biology is the discovery of the regulatory mechanisms governing the expression of genes in the cell. The expression of a gene is controlled by many mechanisms. A key junction in these mechanisms is mRNA transcription regulation by various proteins, known as *transcription factors* (TFs), that bind to specific sites in the promoter region of a gene and activate or inhibit transcription. Loosely speaking, we can view the promoter region as an encoding of a "program," whose "execution" leads to the expression of different genes at different points in time and in different situations. To a first-order approximation, this "program" is encoded by the presence or absence of TF binding sites within the promoter. In this paper, we attempt to construct a unified model that relates the promoter sequence to the expression of genes, as measured by DNA microarrays.

There have been several attempts to relate promoter sequence data and expression data. Broadly, these can be classified as being of one of two types. Approaches of the first and more common type use gene expression measurements to define groups of genes that are potentially co-regulated. They then attempt to identify regulatory elements by searching for commonality (e.g., a commonly occurring motif) in the promoter regions of the genes in the group (see for example [5, 21, 26, 29, 31]). Approaches of the second type work in the opposite direction. These approaches first reduce the sequence data into some predefined features of the gene, e.g., the presence or absence of various potential TF binding sites (using either an exhaustive approach, say, all DNA-words of length 6-7, or a knowledge-based approach, say, all TRANSFAC [32] sites). They then try and exploit these features as well as the expression data in a combined way. Some build models that characterize the expression profiles of groups or clusters of genes (e.g., [3, 27, 7]). Others attempt to identify combinatorial interactions of transcription factors by scoring expression profiles of groups of genes having a combination of the identified motifs [24].

Unlike the approaches described above, our aim is to build a unified model that spans the entire process, from the raw promoter sequence to the observed genomic expression data. We provide a unified probabilistic framework, that models both parts of the process in a single framework. Our model is oriented around a set of variables that define, for each gene g and transcription factor t, whether t regulates g by binding to g's promoter sequence. These variables are hidden, and a key part of our learning algorithm is to induce their values from the data. The model then contains two components. The first is a model that predicts, based on g's promoter sequence, whether t regulates g (or more precisely, when t is active, whether it can regulate g). The second predicts, based on the regulation events for a particular gene g, its expression profile in different settings.

A key property of our approach is that these two components are part of a single model, and are trained together, to achieve maximum predictiveness. Our algorithm thereby simultaneously dis-

*Contact author

covers motifs that are predictive of gene expression, and discovers clusters of genes whose behavior is well-explained by putative regulation events.

Both components of our model have significant advantages over other comparable approaches. The component that predicts regulation from the promoter sequence uses a novel discriminative approach, that avoids many of the problems associated with modeling of the background sequence distribution. More importantly, the component that predicts mRNA expression from regulation learns a model that identifies combinatorial interactions of regulation events. In yeast cell-cycle data, for example, we might learn that, in the G1 phase of the cell cycle, genes that are regulated by Swi6 and Swi4 but not by Mcm1 are over-expressed.

Finally, our use of a general-purpose probabilistic framework allows us to integrate other sources of information into the same unified model. Of particular interest are the recent experimental assays for localizing binding sites of transcription factors [25, 28]. These attempt to detect directly to which promoter regions a particular TF protein binds *in vivo*. We show how the data from these assays can be integrated seamlessly and coherently into our model, allowing us to tie a specific transcription factor with a common motif in the promoter regions to which it binds.

We demonstrate our results in analysis of yeast cell cycle. We combine the known genomic yeast sequence [8], microarray expression data of Spellman *et al.* [30], and the TF binding localization data for 9 transcription factors that are involved in cell-cycle regulation of Simon *et al.* [28]. We show that our framework discovers overlapping sets of genes that strongly appear to be co-regulated, both their manifestation in the gene expression data and in the existence of highly significant motifs in their promoter region. We also show that this unified model can predict expression directly from promoter sequence. Finally, we present how our algorithm also provides valuable biological insight into the domain, including cyclic behavior of the different regulatory elements, and some interesting combinatorial interactions between them.

2. Model Overview

In this section, we give a high-level description of our unified probabilistic model. In the subsequent sections, we elaborate on the details of its different components, and discuss how the model can be trained as a single unified whole to maximize its ability to predict expression as a function of promoter sequence.

Our model is based on the language of *probabilistic relational models* (PRMs) [20, 12]. For lack of space, we do not review the general PRM framework, but focus on the details of the model, which follows the application of PRMs to gene expression by Segal *et al.* [27]. A simplified version of our model is presented in Fig. 1(a). We now describe each element of the model.

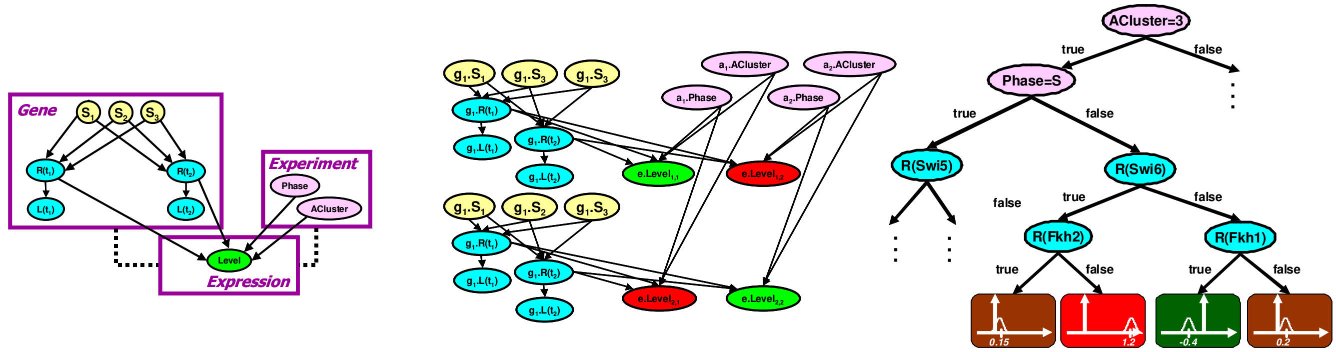
The PRM framework represents the domain in terms of the different interacting biological entities. In particular, we have an *object* for every gene g. Each gene object is associated with several *attributes* that characterize it. Most simply, each gene has attributes $g.S_1, \ldots, g.S_n$ that represent the base pairs in its hypothesized promoter sequence. For example, we might have $g.S_1 = A$. More interestingly, for every transcription factor (TF) t, a gene has a *regulation variable* $R(t)$, whose value is *true* if t binds somewhere within g's promoter region, indicating regulation (of some type) of g by t. The regulation variables depend directly on the gene's promoter sequence, with each TF having its own model, as described in Section 3.1. Note that the regulation variables are hidden in the data; in fact, an important part of our task is to infer their values.

In addition, as we mentioned, our approach allows the incorporation of data from binding localization assays, which attempt to measure the extent to which a particular transcription factor protein binds to a gene's promoter region. This measurement, however, is quite noisy, and it provides, at best, an indication as to whether binding has taken place. One can ascribe regulation only to those measurements where a statistical significance test indicates a very strong likelihood that binding actually took place [28], but it is then misleading to infer that binding did not take place elsewhere. Our framework provides a natural solution to this problem, where we take the actual regulation variables to be hidden, but use localization measurements as a noisy indicator of the actual regulation event. More precisely, each gene g also has a localization variable $L(t)$ for each TF t, which indicates the value of the statistical test for the binding assay for t and g. Our model for the values of this variable clearly depends on whether t actually regulates g; for example, values associated with high-confidence binding are much more likely if $g.R(t)$ takes the value *true*. We describe the model in detail in Section 3.2.

The second main component of our model is the description of expression data. Thus, in addition to gene objects, we also have an object a for every array, and an object e for every expression measurement. Each expression e is associated with a gene $e.Gene = g$, an array $e.Array = a$, and a real-valued attribute $e.Level$, denoting the mRNA expression level of the gene g in the array a. Arrays also have attributes; for example, each array a might be annotated with the cell-cycle phase at the point the experiment was performed, denoted by $a.Phase$. As the array attributes are not usually sufficient to explain the variability in the expression measurements, we often also introduce an additional hidden variable $a.ACluster$ for each array, which can capture other aspects of the array, allowing the algorithm both to explain the expression data better, and to generate more coherent and biologically relevant clusters of genes and experimental conditions.

Our model defines a probability distribution over each gene g's expression level in each array a as a (stochastic) function of both the different TFs that regulate g and of the properties of the specific experiment used to produce the array a. Thus, we have a model that predicts $e.Level$ as a (stochastic) function of the values of its *parents* $g.R(t)$ and $a.Phase$ (where $g = e.Gene$ and $a = e.Array$). As we discuss in Section 3.3, our model for expression level allows for combinatorial interactions between regulation events, as well as regulation that varies according to context, e.g., the cell-cycle phase.

The model that we learn has a very compact description. As we discuss below, we learn one *position specific scoring matrix (PSSM)* for each TF t, which is then used to predict $g.R(t)$ from the promoter sequence of g for all genes g. Similarly, we learn a single model for $e.Level$ as a function of its parents, which is then applied to all expression measurements in our data set. However, the instantiation of the model to a data set is quite large. In a specific instantiation of the PRM model we might have 1000 gene objects, each with 1000 base pairs in its promoter region. We might be interested in modeling 9 TFs, and each gene would have a regulation variable for each of them. Thus, this specific instantiation would contain 9000 regulation variables. Our gene expression dataset might have 100 arrays, so that we have as many as 1000×100 expression objects (if no expressions are missing). Thus, an instantiation of our model to a particular dataset can contain a large number of objects and variables that interact probabilistically with each other. The resulting probabilistic model is a *Bayesian network* [22], where the local probability models governing the behavior of nodes of the same type (e.g., all nodes $g.R(t_1)$ for different genes g) are shared. Fig. 1(b) contains a small instantiation of such as network, for two genes with promoter sequence of length 3, two TFs, and two arrays.



(a) (b) (c)

Figure 1: (a) PRM for the unified model. (b) An instantiation of the PRM to a particular dataset with 2 genes each with a promoter sequence of length 3, 2 TFs, and 2 arrays. (c) An example tree-CPD for the e.*Level* attribute in terms of attributes of e.*Gene* and e.*Array*.

3. A Unified Probabilistic Model

In this section, we provide a more detailed description of our unified probabilistic model, as outlined above. Specifically, we describe the probabilistic models governing: the regulation variables $g.R(t)$; the localization variables $g.L(t)$; and the expression level variables $e.Level$. In the next section, we discuss how the model as a whole can be learned from raw data.

3.1 Model for Sequence Motifs

The first part of our model relates the promoter region sequence data to the *Regulates* variables. Experimental biology has shown that transcription factors bind to relatively short sequences, and that there can be some variability in the binding site sequences. Thus, most standard approaches to uncovering transcription factor binding sites, e.g., [1, 26, 29], search for relatively short sequence motifs in the bound promoter sequences.

A common way of representing the variability within the binding site is by using a *position specific scoring matrix* (PSSM). Suppose we are searching for motifs of length k (or less). A PSSM $\vec{w}$ is a $k \times 4$ matrix, assigning for each position $i = 1, \ldots, k$ and letter $l \in \{A, C, G, T\}$ a weight $w_i[l]$. We can then score a putative k-mer binding site $\boldsymbol{S} = s_1, \ldots, s_k$ by computing $\text{PSSM}(\boldsymbol{S}, \vec{w}) = \sum_i w_i[s_i]$.

The question is how to learn these PSSM weights. We start by defining the model more carefully. Recall that, our model associates, with each gene g, a binary variable $g.R(t) \in \{\mathit{true}, \mathit{false}\}$ which denotes whether the TF t regulates the gene or not. To simplify notation in this subsection, we focus attention on the regulation for a particular TF t, and drop the explicit reference to t from our notation. Furthermore, recall that each gene g has a promoter sequence $g.S_1, \ldots, g.S_n$, where each $S_i \in \{A, C, G, T\}$.

The standard approaches to learning PSSM is by training a probabilistic model of binding sites that maximizes the likelihood of sequences (given the assignment of the regulates variables) [1, 26]. These approaches rely on a clear probabilistic semantics of PSSM scores. We denote by θ_0 the probability distribution over nucleotides according to the background model. For simplicity, we use a Markov process of order 0 for the background distribution. (As we will see, the choice of background model is not crucial in the discriminative model we develop.) We use ψ_j to denote the distribution of characters in the jth position of the binding site. The model then assumes that if t regulates g, then g's promoter sequence has the background distribution for every position in the promoter sequence, except for a specific k-mer, $i+1, \ldots, i+k$, where t binds to it. If t does not regulate g, we have the background distribution for all positions in the sequence. Then,

$$\begin{array}{rcl} P(S_1, \ldots, S_n \mid g.R = \mathit{false}) & = & \prod_\ell \theta_0[S_\ell] \\ P(S_1, \ldots, S_n \mid g.R = \mathit{true}) & = & \\ \multicolumn{3}{r}{\prod_\ell \theta_0[S_\ell] \sum_j \frac{1}{n-k} \prod_{i=1}^k \frac{\psi_i[S_{i+j}]}{\theta_0[S_{i+j}]}} \end{array} \qquad (1)$$

where we assume a uniform prior over the binding position in case of regulation.[1]

The probabilistic approaches train the parameters $\psi_i[l]$ to maximize the probability of the training sequence. Once these parameters are found, they set the PSSM weights to $w_i[l] = \log \frac{\psi_i[l]}{\theta_0[l]}$. Such approaches are *generative*, in the sense that they try to build a model of the promoter region sequence, and training succeeds when the model gives the given sequences high probability. However, these approaches can often be confused by repetitive motifs that occur in many promoter sequences. These motifs have to be filtered out by using an appropriate background distribution [31].

We approach this problem from a different perspective. Recall that our aim is to model the dependence of the gene's genomic expression profile on its promoter sequence. For this task, we do *not* need to model the sequence; we need only estimate the probability that the transcription factor regulates the gene given the promoter region. Thus, it suffices to find motifs that *discriminate* between promoter regions where the transcription factor binds and those where it does not. As we show, this more directed goal allows us to avoid the problem of learning background distribution of promoters and to focus on the classification task at hand.

More formally, we are only interested in the conditional probability of $g.R$ given the sequence $S_1, \ldots, S_n$. If we have a model of the form of (1), and apply Bayes rule, we obtain:

$$P(g.R = \mathit{true} \mid S_1, \ldots, S_n) = \mathit{logit}(x)$$

where $\mathit{logit}(x) = \frac{1}{1+e^{-x}}$ is the logistic function. and

$$\begin{array}{rcl} x & = & \log \dfrac{P(S_1, \ldots, S_n, g.R = \mathit{true})}{P(S_1, \ldots, S_n, g.R = \mathit{false})} \\ & = & \log \left(\dfrac{P(g.R = \mathit{true})}{P(g.R = \mathit{false})} \dfrac{1}{n-k} \displaystyle\sum_j \prod_{i=1}^k \dfrac{\psi_i[S_{i+j}]}{\theta_0[S_{i+j}]} \right) \end{array}$$

Where $P(R = \mathit{true})$ is the prior on binding occurrence.

[1] Some methods, such as MEME [1], relax the assumption of a single binding site. For lack of space, we omit this extension here.

For the goal of predicting the probability of $g.R$ given the sequence, the background probabilities are irrelevant as separate parameters. Instead, we can parameterize the model (2) simply using k position-specific weights $w_j[l]$ and a threshold $v = \log \frac{P(g.R=\mathit{true})}{P(g.R=\mathit{false})}$. Thus, we write

$$P(g.R = \mathit{true} \mid S_1, \ldots, S_n) = \mathit{logit}\left(\log\left(\frac{v}{n-k}\sum_j \exp\{\sum_i w_i[S_{i+j}]\}\right)\right) \quad (2)$$

As we discuss in Section 4.1, we can train these parameters directly, so as to best predict $g.R$.

3.2 Localization Model

Localization measurements, when available, provide strong evidence about regulation relationships. To integrate this data into our model, we need to understand the nature of the localization assay. Roughly speaking, these experiments measure the ratio of "hits" for each DNA fragment between a control set of DNA extracted from cells, and DNA that was filtered for binding to the TF of interest. This assay is noisy, and thus we cannot simply "read off" binding. Instead, the experimental protocol [25] uses a statistical model to assign a *p-value* to various ratios. A ratio with a small p-value suggests significant evidence for binding by the TF. Larger p-values can indicate a weaker binding or experimental noise.

A naive approach would be to assert simply that binding takes place if the p-value is below some threshold, and does not occur otherwise. This approach, however, is naive, first because even small p-values do not guarantee binding, but most importantly, because somewhat larger p-values that are above our threshold, although not definitive, might still be suggestive of binding.

Thus, a more appropriate model is to treat the localization experiment as a *noisy sensor* of the *Regulates* variables $g.R(t)$. More precisely, for each TF t for which we have localization measurements, we introduce a new variable $L(t)$, such that $g.L(t)$ represents the localization evidence regarding the binding of t to g. The value of $g.L(t)$ is the p-value computed in the experimental assay.

It remains to determine how to model the connection between the "sensor" $g.L(t)$ and the actual event $g.R(t)$. In our probabilistic framework, we can simply introduce a probabilistic dependency of $g.L(t)$ on $g.R(t)$, using $P(g.L(t) \mid g.R(t))$ to specify our model of this interaction. There are two cases. If $g.R(t) = \mathit{false}$, we expect $g.L(t)$ to be determined by the noise in the assay. By design, the statistical procedure used is such that $P(g.L(t) \mid g.R(t) = \mathit{false})$ has a uniform distribution on $[0, 1]$ (this is exactly the definition of the p-value here). If $g.R(t) = \mathit{true}$, then we expect $g.L(t)$ to be small. We choose to model this using a density $p(g.L(t) = \rho \mid g.R(t) = \mathit{true}) = c\exp(-w\rho)$, the exponential distribution with weight w, where $c = w/(1-\exp(-w))$ is a normalization constant ensuring that the density function integrates to 1. Based on examination of p-values in the experiments of Simon *et al.*, we choose $w = 20$ in our experiments. Now, once we observe $g.L(t)$, the probability of this observation propagates to $g.R(t)$. If $g.L(t)$ is very small, then it is more likely that it was generated from $g.R(t) = \mathit{true}$. If it is larger, then it is more probable that it was generated from $g.R(t) = \mathit{false}$. This model allows us to use the location-specific binding data as guidance for inferring regulation relationships, without making overly strong assumptions about their accuracy.

3.3 Model for Gene Expression

We now consider the second major component of our unified model: the dependence of the gene's expression profile on the transcriptional regulation mechanisms. More precisely, our model specifies how, in different experimental conditions, various TFs combine to cause up-regulation or down-regulation of a gene. From a technical perspective, we need to represent a predictive model for $e.\mathit{Level}$ based on the attributes of the corresponding gene g and array a. There are many possible choices of predictive models. One obvious choice is linear regression, where we hypothesize that the expression level is normally distributed around a value, whose mean is a linear function of the presence or absence of the different attributes (similar to Bussemaker *et al.* [7]). However, this approach is limited in two ways. First, one has to provide special treatment for attributes whose value space is nominal but not binary, e.g., the cell cycle phase. A far more fundamental limitation is that a linear regression model captures only linear interactions between the attributes, while it is well-known that the interactions between TF that lead to activation or inhibition are much more complex, and combinatorial in nature.

Following our earlier work [27], we choose to use the framework of tree-structured conditional distributions [4, 13], a formalism closely related to decision trees. This representation is attractive in this setting since it can capture multiple types of combinatorial interactions and context specific effects.

Formally, a *tree-structured CPD* $\mathcal{T}$ for a variable X given some set of attributes $V_1, \ldots, V_n$ is a rooted tree; each *node* in the tree is either a *leaf* or an *interior node*. Each interior node is labeled with a test of the form $V = v$, for $V \in \{V_1, \ldots, V_n\}$ and v one of its values. Each of these interior nodes has two outgoing *arcs* to its children, corresponding to the outcomes of the test (true or false). Each of the leaves is associated with a distribution over the possible values of X. In our case, X is the expression level, and therefore takes real values. We therefore associate with each leaf ℓ a univariate Gaussian distribution, parameterized by a mean μ_ℓ and variance σ^2_ℓ. Fig. 1(c) shows an example of a partial tree-CPD. We use $\mathcal{S}_\mathcal{T}$ to denote the qualitative structure of the tree, and $\theta_\mathcal{T}$ to denote the parameters at the leaves.

In our domain, the tests in the tree-CPD are about attributes of the gene, (e.g., $g.R(\mathit{Swi6}) = \mathit{true}$) or attributes of the array (e.g., $a.\mathit{Phase} = \mathit{S}$). Each leaf corresponds to a *grouping* of measurements. This is a "rectangle" in the expression data that contains the expression levels of a subset of genes (defined by the tests on the gene attributes) in a subset of the arrays (defined by the tests on array attributes).

It is important to realize that the tree-CPD representation can encode combinatorial interactions between different TFs. For example, as shown in the figure, in arrays in cluster 3, except for those in phase S of the cell cycle, genes that are regulated by Swi6 and not Fkh2 are highly overexpressed, whereas those that are also regulated by Fkh2 are only very slightly overexpressed. In addition, the tree can model context specific interactions, where different attributes predict the expression level in different branches of the tree. In our domain, this can capture variation of regulation mechanisms across different experimental conditions. For example, in phase S of the cell cycle, the set of relevant TFs is completely different.

4. Learning the model

In the previous section, we described the different components of our unified probabilistic model. In this section, we consider how we learn this model from data: promoter sequence data, genomic expression data, and (if available) localization data. A critical part of our approach is that our algorithm does not learn each part of the model in isolation. Rather, our model is trained as a unified whole, allowing information and (probabilistic) conclusions from one type of data to propagate and influence our conclusions about another type. The key to this joint training of the model are the regulation variables, that are common to the different components. It is important to remember that these variables are hidden, and part of the task of the learning algorithm is to hypothesize their values.

There are several nontrivial subtasks that our learning algorithm must deal with. First, as we discussed, we need to learn the parameters of the discriminative motif models, as described in Section 3.1. Second, we need to learn both the qualitative tree structure and the parameters for our tree CPD. Finally, we need to deal with the fact that our model contains several hidden variables: the different *Regulates* variables, and the array cluster variables. We discuss each of these subtasks in turn.

4.1 Learning the Sequence Model

Our goal in this section is to learn a model for the binding sites of a TF t that predicts well whether t regulates a gene g by binding somewhere in its promoter region. We defer the treatment of hidden variables to Section 4.3; hence, we assume, for the moment, that we are given, as training data, a set of genes $g[1], \ldots, g[M]$ and their promoter sequences, and that we are told, for each gene $g[m]$, whether t regulates $g[m]$ or not.

Given M genes $g[1], \ldots, g[M]$, and the values of their regulation variable $g[m].R(t)$, we try to maximize the conditional log probability

$$\sum_m \log P(g[m].R(t) \mid g[m].S_1, \ldots, g[m].S_n).$$

Our task therefore is to find the values of the parameters $w_j[c]$ and v for the PSSM of t that maximize this scoring function. It is easy to see that this optimization problem has no closed form solution, and that there are many local maxima. We therefore use a conjugate gradient ascent, to find a local optimum in the parameter space.

Conjugate gradient starts from an initial guess of the weights $\vec{w}^{(0)}$. As for all local hill climbing methods, the quality of the starting point has a huge impact on the quality of the local optimum found by the algorithm. In principle, we can use any motif learning algorithm to initialize the model. We use the method of Barash and Friedman [2], which efficiently scores many motif "signatures" for significant over-abundance in the promoter sequences which the TF supposedly regulates as compared to those it does not. It uses the random projection approach of Buhler and Tompa [6] to generate motif seeds of length 6–15, and then scores them using the hypergeometric significance test. As described by Barash and Friedman, this approach can efficiently detect potential initialization points that are discriminative in nature. Each seed produced by this method is then expanded to produce a PSSM of the desired length, whose weights serve as an initialization point for the conjugate gradient procedure.

4.2 Learning the Expression Model

A second subtask is to learn the model associated with the expression data, i.e., the model of $e.$*Level* as a function of gene and array attributes: the regulation variables, the array cluster variables, and any other attributes we might have (e.g., the array cell-cycle phase). Again, we temporarily assume that these attributes are all observed, deferring the treatment of hidden variables to Section 4.3.

As we discussed, we use a tree-structured probabilistic model for $e.$*Level*, with a Gaussian distribution at each leaf; hence, our task is to learn both the qualitative structure of the tree and the parameters for the Gaussians. Our approach is based on the methods used by Segal *et al.* [27]; we briefly review the high-level details. There are two issues that need to be addressed: a *scoring function*, used to evaluate the "goodness" of different candidate structures relative to the data, and a *search algorithm* that finds a structure with a high score among the superexponentially many structures.

We use *Bayesian model selection* techniques to score candidate structures [16, 12] . The *Bayesian score* of a structure is defined as the *posterior* probability of the structure $\mathcal{S}$ given a data set D:

$$P(\mathcal{S} \mid D) \propto P(\mathcal{S}) \int P(D \mid \mathcal{S}, \theta) P(\theta \mid \mathcal{S}) d\theta$$

where $P(\mathcal{S})$ is the prior over structures, and $P(\theta \mid \mathcal{S})$ is the prior over parameter values for each structure. This expression evaluates the fit of the model to the data by averaging the likelihood of the data over all possible parameterizations of the model. This averaging regularizes the score and avoids overfitting the data with complex models. When the training data is fully observed, and the likelihood function and parameter prior come from certain families, Bayesian score often has a simple analytic form [16], as a function of the sufficient statistics of that model.

In our case, the set of possible structures are the tree structures $\mathcal{S}_\mathcal{T}$. Each parameterization $\theta_\mathcal{T}$ for the leaves of the tree defines a conditional distribution over X given $V_1, \ldots, V_n$. Given a data set $D = \{V_1[m], \ldots, V_n[m], X[m]\}_{m=1}^M$, we want to find the structure $\mathcal{S}_\mathcal{T}$ that maximizes

$$\int \prod_m P(X[m] \mid V_1[m], \ldots, V_n[m], \mathcal{S}_\mathcal{T}, \theta_\mathcal{T}) P(\theta_\mathcal{T} \mid \mathcal{S}_\mathcal{T}) d\theta_\mathcal{T}.$$

If we choose an independent *normal-gamma* prior over the Gaussian parameters at each leaf in the tree, this integral has a simple closed form solution; see [11, 17] for details.

Having defined a metric for evaluating different models, we need to search the space of possible models for one that has a high score. As is standard in both CPD-tree and Bayesian network learning [9, 13, 16], we use a greedy local search procedure that maintains a "current" candidate structure and iteratively modifies it to increase the score. At each iteration, we consider a set of simple local transformations to the current structure, score all of them, and pick the one with the highest score. The two operators we use are: *split* — replaces a leaf in a CPD-tree by an internal node, and labels it with some binary test $V = v$; and *trim* — replaces the entire subtree at an internal node by a single leaf. To avoid local maxima, we use a variant of simulated annealing: Rather than always taking the highest scoring move in each search step, we take a random step with some probability, which decays exponentially as the search progresses.

4.3 Dealing with Hidden Variables

In this section, we present our overall learning algorithm, which learns a single model that predicts expression from promoter sequence. In a sense, our algorithm "puts together" the two learning algorithms described earlier in this section. Indeed, if the regulation (and cluster) variables were actually observed in the data, then that is all we would need to do: simply run both algorithms separately. Of course, these variables are not observed; indeed, inferring their values is an important part of our goal. Thus, we now have to learn a model in the presence of a large number of hidden variables — $g.R(t)$ for every g, t, as well as $a.$*ACluster* for every a.

The main technique we use to address this issue is the *expectation maximization* (EM) algorithm [16]. However, in applying EM in this setting, we have to deal with the large scale of our model. As we discussed, although our PRM model of Fig. 1(a) is compact, it induces a complex set of interactions. In the experiments we describe below, we have 795 genes, 9 TFs, and 77 arrays, resulting in a Bayesian network model with 7242 hidden variables. Moreover, the nature of the data is such that we cannot treat genes (or arrays) as independent samples. Instead, any two hidden variables are dependent on each other given the observations (see Friedman *et al.* [12] for an elaboration of this point). A key simplification is based on the observation that the two parts of the model — the TF model and the expression model — decouple nicely, allowing us to deal with each separately, with only limited interaction via the *Regulates* variables.

We begin by learning the expression submodel. A good initialization is critical to avoid local maxima. Hence, we initialize the *Regulates* variables with some reasonable starting point. Possibilities include: direct inference from localization data; a verified source of TFs such as the TRANSFAC [32] repository; or the results of applying a motif-discovery algorithm to the results of an expression clustering algorithm (as in [5, 26, 29, 31]). We also initialize the *ACluster* attributes using the output of some standard clustering program. Treating these initial values as fixed, we then proceed with the first iteration of learning the expression model.

Using this hard assignment to all of the variables, we begin by learning a tree-CPD, as described in Section 4.2. We then fix the tree structure, and run EM on the model to adapt the parameters. As usual, EM consists of: an E-step, where we run inference on the model to obtain a distribution over the values of the hidden variables; and an M-step, where we take the distribution obtained in the E-step and use the resulting soft assignment to each of the hidden variables to re-estimate the parameters using standard maximum likelihood estimation. However, computational reasons prevent us from executing this process simultaneously for all of the hidden variables — $g.R(t)$ and $a.\mathit{ACluster}$. We therefore perform an incremental EM update, treating the variables a group at a time. We leave most of the variables with a hard assignment of values; we "soften" the assignment to a subset of the variables, and adapt the model associated with them; we then compute a new hard assignment to this subset, and continue.

More precisely, we iterate over TFs t, for each one performing the following process:

- We "hide" the hard assignment to the values of the variables $g.R(t)$ for all g, leaving the other variables ($g.R(t')$ for $t' \neq t$, and $a.\mathit{ACluster}$) with their current hard assignment values.
- We perform an E-step, running inference on this model to obtain a posterior distribution for each variable $g.R(t)$. Note that, since $g.R(t)$ is hidden and is part of both the PSSM model and the expression model, inference is done jointly on both, and the posterior of $g.R(t)$ is determined by their combined probabilistic influence on $g.R(t)$ as propagated through the network. We also note that, with fixed assignments to all the other variables, the different $g.R(t)$ variables are all conditionally independent, so that this inference can be done exactly and very efficiently.
- We perform an M-step, adapting the parameters in the model appropriately. Specifically, the M-step may change the Gaussian parameters at the leaves in the tree-CPD, because genes now "end up" at different leaves in the tree, based on their new distribution for $g.R(t)$. Moreover, the M-step involves updating the parameters of the PSSM model. As discussed in Section 4.1, there is no closed form solution for performing this update and we thus use conjugate gradient ascent, replacing the hard classification for $g.R(t)$ which we assumed exists in Section 4.1, with a soft classification for $g.R(t)$, equal to the posterior probability of $g.R(t)$ as computed in the E-step.
- We pick a new hard assignment for $g.R(t)$ for every g, by choosing its most likely value from the distribution.

This process is executed in a round robin for every TF. A similar process is executed for the variables $a.\mathit{ACluster}$. Once we finish these EM updates, we either terminate, or repeat the whole sequence using the updated assignment to the hidden variables: learning the expression model, followed by EM updates. This process repeats until convergence.

From an intuitive perspective, our approach is executing a very natural process. As we mentioned, had the $g.R(t)$ variables been observed, we could have trained each part of the model separately. In our setting, we allow the expression data and the sequence data to simultaneously influence each other and together determine better estimates for the values of $R(t)$.

5. Experimental results

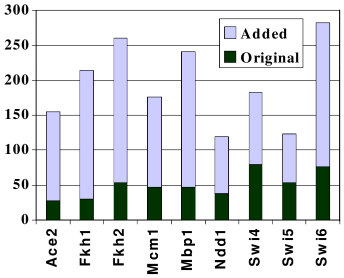
We evaluated our algorithm on a combined data set relating to yeast cell cycle. The data set involved all 795 genes in the cell cycle data set of Spellman *et al.* [30]. For each gene, we had the sequence of the gene's promoter region (1000bp upstream of the ORF), the gene's expression profile on the 77 arrays of [30], and the localization data of Simon *et al.* [28] for nine transcription factors: Fkh1, Fkh2, Swi4, Swi5, Swi6, Mbp1, Ace2, Ndd1, Mcm1. To simplify the following discussion, we use $\mathit{Reg}_L(t)$ to refer to the set of genes where the statistical analysis of Simon *et al.* indicated binding by the TF t with p-value 0.001 or lower. We use $\mathit{Reg}(t)$ to refer to the set of genes g where our algorithm assigned a posterior probability to $g.R(t)$ which is greater than 0.5.

5.1 Prediction Tests

We began by experimenting with several different models, involving various amounts of learning. To objectively test how each model generalizes to unobserved data, we used 5-fold cross validation. In each run, we trained using 596 genes, and then tested on the expression levels of the remaining 199 held back genes. The partition into training and test set done randomly, and all models used exactly the same partitions. We report the average log-likelihood per gene. Differences in this measure correspond to multiplicative differences in the prediction probability for the test data. For example, an improvement of +1 in the average log-likelihood represents that the expression level predictions have twice the probability in the new model relative to the old one.

The models and results are as follows:

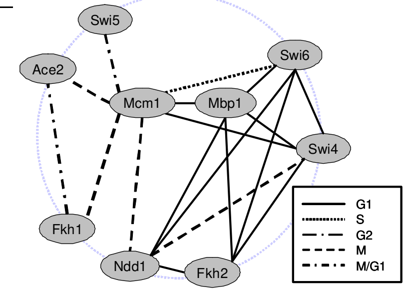
- M_{cg}, where we tried to predict the expression data using only the cell-cycle phase attribute for arrays and a corresponding *G-phase* attribute for genes, which represents the phase at which the gene is most active (see [30, Supp. data]); average log-likelihood: -112.24 ± 11.42 (standard deviation).
- M_{cl}, where we tried to predict the expression data using the cell-cycle phase attribute for arrays and *Regulates* variables for genes, whose values were simply set according to $\mathit{Reg}_L(t)$; average log-likelihood: -134.87 ± 15.29.
- M_{cle}, which is the same as M_{cl}, except that the localization data is treated as a noisy sensor $g.L(t)$ of a hidden *Regulates* variable $g.R(t)$; average log-likelihood: -121.48 ± 11.96.
- M_{cla}, which introduces the hidden $a.\mathit{ACluster}$ attributes (with 5 possible values) into M_{cle}, and trains the model using EM; average log-likelihood: -103.76 ± 5.72.
- M_{cls}, which introduces the sequence data into the model of M_{cla}, giving us our full model; average log-likelihood: -94.59 ± 4.13.



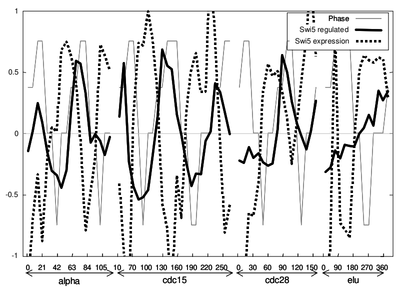
(a)

TF	A	B	C	p-value	PSSM
Ace2	10	9	1	1.4e-06	CCTcGC
Fkh1	29	25	8	4.4e-10	AACAAA
Fkh2	29	29	10	5.4e-11	GGcGGG
Mbp1	65	56	8	1.9e-45	ACGCG
Mcm1	28	24	2	4.2e-18	TTCcTA
Ndd1	17	28	1	1.9e-24	AcCTCG
Swi4	41	37	5	6.4e-26	CGCGAA
Swi5	28	23	2	4.9e-15	AGGCCG
Swi6	50	52	6	2.3e-48	CGcGTc

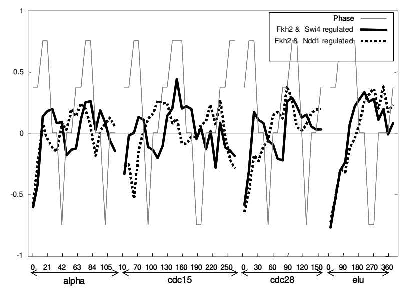
(b)



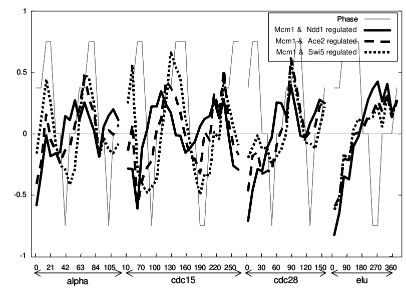
(c)



(d)



(e)



(f)

Figure 2: (a) Changes in classification of *Regulates* variables from the initial to the final iteration. (b) Table showing the PSSMs found in the analysis and summarizing their specificity values. The table reports percent "hits" in 3 groups: A — genes in *Reg*$_L(t)$; B — genes in *Reg*(t); C — not in *Reg*(t). The table also reports specificity p-value for the regulated genes. (c) Figure of putative combinatorial interactions based on the M_{cla} model. (d), (e) & (f) Plots of the average expression for subsets of genes regulated by different combinations of TFs. The x-axis denotes arrays along the four time courses of Spellman *et al.*, and the y-axis denotes average expression level of the genes in each group. The cell cycle phase is shown by by the thin gray line that with peaks in the G1 phase and troughs in the G2 phase.

We see that our baseline model M_{cg} does fairly well, which is not surprising: The *G-phase* attributes were chosen specifically to be an accurate description of the expression profile of the genes. More interesting is the comparison between the other four models. We can see that the localization data alone can explain only a small part of the data, and achieves a fairly poor predictive performance. This is mainly due to the conservative approach of Simon *et al.* [28], who selected the threshold defining *Reg*$_L(t)$ to minimize the number of false positives; thus, most of the genes in our data set had all the *Regulates* attributes set to false.

Treating the localization data as a noisy sensor only and running EM improves the accuracy of the predictions. An examination of the data shows that the EM process substantially changes the values of the *Regulates* variables from their initial values according to *Reg*$_L$, primarily by adding new genes for which regulation is hypothesized, i.e., where the most likely value of $g.R(t)$ is *true*. (We discuss this issue further below.) The new $R(t)$ variables are trained to be much more predictive of the expression data, so it is not surprising that the resulting model achieves a higher score. The addition of the *ACluster* attributes also improves the score substantially, as the five-valued cell cycle phase attribute is not enough to distinguish between qualitatively very different arrays. For example, some of the clusters captures distinctions such as "early" vs. "late" in the time series. These distinctions are important, as later measurements lost synchronization to a certain degree and thus are less "sharp". Finally, most interesting is the fact that, by introducing the sequence, we get a very substantial additional boost in predictive accuracy.

We attempted to isolate the source of this last improvement in accuracy. We tested the full model M_{cls} on the test data, but giving it only the sequence information, rather than both the localization and sequence information, as in our previous experiment. In other words, although the model was learned using localization data, in the test data we are predicting the expression level using *only* the sequence data. In this case, the average log-likelihood is -95.36 ± 3.90, which is almost indistinguishable from the result given the localization data as well. This result is quite important, because it suggests that our model has, indeed, learned to predict the expression data directly from sequence data!

5.2 Inferring Regulation

One of the main factors that contribute to the success of the learning algorithm is its ability to change the classification of the $R(t)$ variables from their original values as determined by localization alone. The algorithm changes these variables to provide a better explanation of the training data. The only constraints on this transformation are those imposed by our choice of the probabilistic model for localization, which makes it very likely that if the binding of t to g was associated with a low p-value, the value of $g.R(t)$ will remain *true*. Indeed, if we examine the genes for which $g.R(t)$ changed its value, we see that the value changed from *true* to *false* for at most 1 or 2 genes per TF. Thus, the procedure mainly introduced new genes into *Reg*(t). Fig. 2(a) compares the original and final number of genes in *Reg*(t). We see that the procedure increased the number of regulated genes for all the TFs. For some TFs (e.g., Ndd1, Swi4), the change was fairly minor; others (e.g., Fkh1, Fkh2, Swi6) increased by close to 5-fold.

There are several explanations for these changes. In some cases, the genes that we added are also truly regulated by the TF, but the signal was not visible in the localization data. For example, our model predicts that Clb1 and Cdc5 are regulated by Ndd1. There is evidence that these genes are regulated by Mcm1 which works together in G2/M with Ndd1. The analysis of Simon *et al.* failed to find binding. Additionally, our model suggests that Fkh1 and Fkh2 are regulated by Fkh1 and not Mcm1. This conclusion fits the recent results of Hollenhorst *et al.* [18].

A second explanation for this phenomenon is the fact that we gave the model only the nine *Regulates* variables in order to try and explain a complex expression matrix. As such, the model sometimes had to "stretch" the boundaries of these variables in order to improve the quality of its predictions. Thus, it is possible that the semantics of $R(t)$ may have changed in order to capture interactions for which no explanation was present in the data. Nevertheless, we believe that the sets $Reg(t)$ are meaningful, and almost certainly co-regulated by some combination of mechanisms.

One strong indicator in favor of this hypothesis is the demonstrated ability of the algorithm to predict expression directly from sequence data, suggesting that there are common features in the promoter region of the genes that our algorithm asserts are co-regulated. We tested this conjecture by looking at the motifs that were discovered by the algorithm. Fig. 2(b) lists, for each TF t, the percentage of genes for which the PSSM learned by our algorithm determined the existence of a motif, i.e., those where $P(g.R(t) = \mathit{true} \mid g.S_1, \ldots, g.S_n) > 0.5$. We compute the percentage in three groups: the genes in $Reg_L(t)$, the genes in $Reg(t)$, and in the remaining genes (those where $g.R(t) = \mathit{false}$). In addition, the table lists the p-value (using a hypergeometric model [2]) of the motif and a pictorial representation of the learned PSSM.

This table shows several trends. We see that some motifs (e.g., Mbp1) appear in a majority of the genes in $Reg(t)$. Moreover, in some cases (e.g., Ace2, Mcm1, Swi5) the motif is very rare in the group of genes for which $g.R(t) = \mathit{false}$. Thus, these motifs are quite specific for the genes they were trained on, as we can see by the p-values. The significance of the p-values suggests that this is not an artifact. In addition, we see that the motifs have a similar concentration in the genes in $Reg_L(t)$. Thus, although these genes are often less than 25% of the training genes, the learned motif is quite common among them. This last result suggests that there is no difference, from the perspective of finding common motifs, between the co-regulated set of genes discovered by Simon *et al.* and those that were introduced into this set by our algorithm.

This conclusion is further validated by comparing the learned PSSMs to the known binding sites in the literature. The PSSMs for Mbp1 and Swi4 are similar to the ones found by Tavazoie *et al.* [31]. The PSSM for Mcm1 is also similar to the one found by Tavazoie *et al.*, except that they discovered a homodimer site that consists of two roughly palindromic parts. Our PSSM for Mcm1 captures only one of these parts. (We note that since our model scans both strands, it suffices for the discriminative model to learn only half of the site.) Our PSSM for Fkh1 matches the model suggested by Zhu *et al.* [33]. On the other hand, our PSSM for Swi5 and Ace2 are quite different from the known ones in the literature.

Finally, it is interesting to examine the PSSMs learned from Ndd1 and Swi6. The current hypothesis for Ndd1 is that it cannot bind to the promoter directly. Rather, it is recruited by either Fkh1/2 or by Mcm1. The PSSM we learned for Ndd1 is somewhat similar to PSSM learned by Simon *et al* for Fkh1. Similarly, Swi6 is recruited by either Swi4 or Mbp1; and, indeed, the PSSM we learn for Swi6 is similar (but not identical) to the published Mbp1 motif.

5.3 Biological analysis

We end the discussion of our results by examining their biological plausibility. First, we tested the extent to which our set of potentially co-regulated genes were actually co-expressed. To do so, we computed, for each array a, the average of the expression levels for the genes in $Reg(t)$. Fig. 2(c) shows the expression of Swi5 and the average expression of Swi5-regulated genes. We see that the expression of Swi5 regulated genes follows a pronounced cyclic behavior that peaks in the M cell cycle phase. We also can see that the Swi5 gene is transcribed before its protein product is being used. This is consistent with knowledge that Swi5 itself is transcriptionally regulated [28, 30]. In addition, the fact that Swi5 is transcribed before the genes it activates fits nicely with biological understanding, where the delay corresponds to the time need for translation of the Swi5 transcript and then the time required for it to bind to its target sites and initiate transcription.

As described above, we expect our model to capture effects of combinatorial regulation. These can be seen when we examine the expression of groups of genes that are regulated by two or more TFs. figreffig:results(e) shows that genes that are regulated by both Fkh2 and Swi4 peak in G1 and late G1, whereas genes regulated by Fkh2 and Ndd1 peak in M or M/G1. This behavior is exactly compatible with our current understanding of the role of these two transcription factors complexes that involve Fkh2. Fig. 2(f) shows the behavior of three complexes involving Mcm1: with Ndd1, Ace2, and Swi5. We can see that genes also co-regulated with Ndd1 peak earlier than the other two. Once again, this behavior is compatible with current biological understanding; see [28].

We then tried to see if we can recover biological insights from our M_{cla} model. Recall that the tree structure learned by our model defines a set of *groupings*, each one defined by one of the leaves of the tree. Each grouping is associated with a list of tests (e.g., $g.R(\mathit{Swi4}) = \mathit{true}$) on attributes of the genes and attributes of the arrays that occur on the path to the leaf. It therefore also corresponds to a "rectangle" in the expression matrix (defined by the genes and the arrays that satisfy the tests), which has a similar expression value. The tests performed along the path indicate which aspects of genes and arrays are important for defining a uniform set of expressions. Thus, they can provide biological understanding of the processes.

However, before imputing biological significance to these tests, it is important to realize that not all of them are truly relevant. While some of these tests are crucial to defining a coherent set of genes and conditions, others might simply be an artifact of our learning algorithm. We therefore performed significance analysis on each of the tests used to produce this grouping, using a t-test to compare the expression measurements in the rectangle with the expression measurements satisfying all other tests defining the grouping except the one in question. We then eliminated tests that appeared irrelevant, remaining with a set of overlapping rectangles, such that all of the tests used to define the rectangle were necessary, with a p-value of less than 5e-4.

We selected groups that were over-expressed, in that their average expression level was greater than 0.5. (Recall that the expression levels of Spellman *et al.* are measured in units of log (base 2) of the ratio to a control.) We note that, in this data, coherent groups are always specific to a particular cell-cycle phase, as determined by the tests in the definition of the group. We then looked for indications of combinatorial regulation: groups which required regulation by two TFs. The resulting "interaction map" is shown in

Fig. 2(c), with an arc between two TFs indicating joint regulation in at least one group. The different arcs indicate joint regulation in different cell-cycle phases.

Many interactions in this map correspond very well with known biology. For example, the interactions between Mbp1 and Swi6, between Swi4 and Swi6, between Ace2 and Mcm1, between Swi5 and Mcm1, between Ndd1 and Mcm1, and between Fkh2 and Mcm1. Other interactions that we would have expected are missing, such as the interaction between Ace2 and Swi5, between Fkh1 and Fkh2, and between Fkh1 and Ndd1. The latter two can perhaps be explained by the fact that Fkh1 and Fkh2 regulate very similar sets of genes, and are therefore somewhat interchangeable. As our learning algorithm looks for compact models that explain the data, it may choose not to introduce one test on a path if if the data is already explained well using some other test. Hence, somewhat redundant tests, such as those on Fkh1 and Fkh2, might never appear together on a path.

Other interactions in the map may suggest potentially interesting hypotheses. For example, finding Ndd1 in interaction with Mbp1 on G1 genes suggests that the Ndd1 protein may participate together with the Forkhead proteins in modulating the expression of Mbp1 targets in G1, as suggested also by the results of Pilpel *et al.* [24]. Other interactions also seem compatible with the results of Pilpel *et al.*, including the interaction between Fkh2 and Swi4, Swi6, and Mpb1.

Finally, we compare the genes in these coherent groups to known annotations of genes from the YPD server [10]. Many of these groups contain a significant portion of genes annotated with a particular functional or cellular role. For example, there is a group of 43 genes that are regulated by Swi6 and Mbp1 but not by Swi4, which contain 8 DNA repair genes (out of 37 such genes in the data). Such a concentration has p-value of $2e-4$. The same group of genes also contain 14/72 chromatin/chromosome structure genes (p-value = $e3-6$), 5/8 DNA polymerase or subunit (p-value = $7e-6$), and 10/36 DNA synthesis genes (p-value = $3e-6$). These results are compatible with our biological understanding about the processes that occur in phase G1 of the cell cycle, when these TFs are active. Another group of 73 genes is defined as being regulated by Swi6, Mbp1, and Fkh2. It contains 17/77 DNA-binding protein genes (p-value = 9e-5) and 20/72 chromatin/chromosome structure genes (p-value = 4e-7). A final example is a set of 72 genes regulated by Ace2 and not by Swi4, containing 11/37 amino-acid metabolism genes (p-value = 8e-5).

6. Discussion

In this paper, we describe a unified probabilistic framework that defines a (simplified) model of the "end-to-end" process of genomic expression: from transcriptional regulation, based on the binding of transcription factors to the gene's promoter region, to the expression data itself. We show how to learn a coherent model based on heterogeneous data: sequence data, expression data, and binding localization data. We demonstrate our algorithm on the yeast cell cycle process, showing that our framework does learn to predict expression from sequence. Our algorithm also finds highly significant motifs in clusters that it asserts are co-regulated, providing a strong biological basis for this claim. We also show that the learned model provides valuable biological insight into the domain, including information about combinatorial regulation by complexes of transcription factors.

Our paper is not the first to try and provide a unified probabilistic framework for these multiple sources of data. Holmes and Bruno [19] describe a simple *Naive Bayes* model for promoter sequence and expression, where the genes are partitioned into disjoint clusters using a hidden *Cluster* variable, whose value probabilistically determines both the sequence (and presence of motifs) of the promoter region and the expression profile. This model is much simpler than ours, and fails to capture important aspects such as combinatorial effects between transcription factors, or the effect of array properties on the expression data. The recent work of Hartemink *et al.* [15] tries to provide a unified framework for localization and expression data. Their approach is based on the Bayesian network based framework for pathway discovery [14, 23]. They use the localization data to guide the discovery by limiting the set of models they consider. Their approach, however, only makes use of the small set of regulation predictions that received very low p-values in the analysis of the localization data. As our results show, it is possible as well as beneficial to make use of all of the localization results. Finally, we note that neither of these approaches, nor any of the others discussed in the introduction, is capable of making use of all three types of data — expression, localization, and sequence — within the context of a single framework.

There are many obvious extensions to this work, which we plan to pursue. It seems fairly clear that the regulation events of a small subset of transcription factors are not sufficient to explain the variability in the expression measurements. For example, it might be that t plays a role both as an activator of some genes, and repressor of others. Thus, learning that $g.R(t)$ holds is not sufficient for predicting the expression of g. It is possible to extend the algorithm by introducing new hidden attributes of the genes. These variables, might allow us to make important distinctions among genes regulated by the same TF, and consequently make better prediction of the expression of these genes. If learned in a guided way, these hidden variables might even correspond to regulation by new regulatory elements or complexes, with associated sequence motifs on the one hand and predictive ability for expressions on the other. In addition, our current PSSM models of binding sites are quite simplistic. As we show, discriminative training allow us to recognize binding sites using such simple models in a specific manner. Nonetheless, these binding sites do not explain all the regulation attributes. To get a better model may require learning more elaborate binding sites models, or explicitly modeling additional attributes of binding sites (their location in the promoter, their relative affinity, etc.) that clearly play a role in the biological system.

Finally, this paper is a step (following [27]) in a long-term project that aims at integrating many different types of data and providing a mechanism for learning a unified probabilistic framework for key genomic processes.

Acknowledgements. We thank Dana Pe'er and Tommy Kaplan for useful discussions. This work was supported by NSF Grant ACI-0082554 under the NSF ITR program, and by the Sloan Foundation. Eran Segal was also supported by a Stanford Graduate Fellowship (SGF). Nir Friedman and Yoseph Barash were supported in part by the Israel Science Foundation (ISF), and the Israeli Ministry of Science. N. Friedman was also supported by an Alon fellowship and the Harry & Abe Sherman Senior Lectureship in Computer Science.

7. REFERENCES

[1] T.L. Bailey and Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 2, pages 28–36. 1994.

[2] Y. Barash, G. Bejerano, and N. Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. In O. Gascuel and B. M. E.

Moret, editors, *Algorithms in Bioinformatics: Proc. First International Workshop*, number 2149 in LNCS, pages 278–293. 2001.

[3] Y. Barash and N. Friedman. Context-specific Bayesian clustering for gene expression data. In *Fifth Annual International Conference on Computational Molecular Biology*. 2001.

[4] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proc. Twelfth Conference on Uncertainty in Artificial Intelligence (UAI '96)*, pages 115–123. 1996.

[5] A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res.*, 8:1202–15, 1998.

[6] J. Buhler and M. Tompa. Finding motifs using random projections. In *RECOMB'01*. 2001.

[7] H.J. Bussemaker, H. Li, and E.D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27:167–71, 2001.

[8] J. M. Cherry, C. Ball, K. Dolinski, S. Dwight, M. Harris, J. C. Matese, G. Sherlock, G. Binkley, H. Jin, S. Weng, and D. Botstein. Saccharomyces genome database. http://genome-www.stanford.edu/Saccharomyces/, 2001.

[9] D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *Proc. Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI '97)*, pages 80–89, 1997.

[10] M.C. Costanzo, M.E. Crawford, J.E. Hirschman, J.E. Kranz, P. Olsen, L.S. Robertson, M.S. Skrzypek, B.R. Braun, K.L. Hopkins, P. Kondu, C. Lengieza, J.E. Lew-Smith, M. Tillberg, and J.I. Garrels. Ypd, pombepd, and wormpd: model organism volumes of the bioknowledge library, an integrated resource for protein information. *Nuc. Acids Res.*, 29:75–9, 2001.

[11] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

[12] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI '99*. 1999.

[13] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 421–460. Kluwer, Dordrecht, Netherlands, 1998.

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

[15] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, and R.A. Young. Combining location and expression data for principled discovery of genetic regulatory network models. In *Pac. Symp. Biocomp. 7*, 2002.

[16] D. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, Dordrecht, Netherlands, 1998.

[17] D. Heckerman and D. Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. In *UAI '95*, pages 274–284. 1995.

[18] P.C. Hollenhorst, G. Pietz, and C.A. Fox. Mechanisms controlling differential promoter-occupancy by the yeast forkhead proteins fkh1p and fkh2p: implications for regulating the cell cycle and differentiation. *Genes Dev.*, 15:2445–56, 2001.

[19] I. Holmes and W. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In *ISMB'00*. 2000.

[20] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *AAAI '98*, 1998.

[21] X Liu, DL Brutlag, and JS Liu. Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. In *Pac. Symp. Biocomput.*, pages 127–38. 2001.

[22] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[23] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215–24, 2001.

[24] Y. Pilpel, P. Sudarsanam, and G.M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29:153–9, 2001.

[25] B. Ren, F. Robert, J.J. Wyrick, O. Aparicio, E.G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T.L. Volkert, C.J. Wilson, S.P. Bell, and R.A. Young. Genome-wide location and function of dna binding proteins. *Science*, 290:2306–9, 2000.

[26] F.P. Roth, P.W. Hughes, J.D. Estep, and G.M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, 16:939–945, 1998.

[27] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17(Suppl 1):S243–52, 2001.

[28] I. Simon, J. Barnett, N. Hannett, C.T. Harbison, N.J. Rinaldi, T.L. Volkert, J.J. Wyrick, J. Zeitlinger, D.K. Gifford, T.S. Jaakkola, and R.A. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.

[29] S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 8, pages 344–54. 2000.

[30] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–97, 1998.

[31] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–5, 1999. Comment in: Nat Genet 1999 Jul;22(3):213-5.

[32] E. Wingender, X. Chen, Fricke E., R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhauser, M. Pruss, F. Schacherer, S. Thiele, and S. Urbach. The TRANSFAC system on gene expression regulation. *Nuc. Acids Res.*, 29:281–283, 2001.

[33] G. Zhu, P. T. Spellman, T. Volpe, P. O. Brown, D. Botstein, T. N. Davis, and B. Futcher. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature*, 406:90–4, 2000.