A Discriminative Model for Identifying Spatial cis-Regulatory Modules

Eran Segal* Roded Sharan†

ABSTRACT

Transcriptional regulation is mediated by the coordinated binding of transcription factors to the upstream region of genes. In higher eukaryotes, the binding sites of cooperating transcription factors are organized into short sequence units, called cis-regulatory modules. In this paper we propose a method for identifying modules of transcription factor binding sites in a set of co-regulated genes, using only the raw sequence data as input. Our method is based on a novel probabilistic model that describes the mechanism of cis-regulation, including the binding sites of cooperating transcription factors, the organization of these binding sites into short sequence modules, and the regulation of a gene by its modules. We show that our method is successful in discovering planted modules in simulated data and known modules in yeast. More importantly, we applied our method to a large collection of human gene sets, and found 83 significant cis-regulatory modules, which included 36 known motifs and many novel ones. Thus, our results provide one of the first comprehensive compendiums of putative cis-regulatory modules in human.

Categories and Subject Descriptors

J.3 Computer applications – life and medical sciences.

General Terms

Algorithms, experimentation.

Keywords

Cis-regulatory module, probabilistic model, transcriptional regulation.

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

†International Computer Science Institute, 1947 Center St., Berkeley, CA 94704. roded@icsi.berkeley.edu.

RECOMB'04, March 27–31, 2004, San Diego, California, USA.

1. INTRODUCTION

Many of the functions carried out by a living cell require the coordination of gene expression, to ensure that genes are expressed when they are needed. To understand biological processes, it is thus necessary to understand this transcriptional network. Much of the information that determines when and where genes are expressed is encoded in an organism's genome sequence. Although we now have sequences for many organisms, our understanding of how this cis-regulatory information is encoded is very limited.

In higher eukaryotes, cis-regulatory information is organized into modular units, called *cis-regulatory modules (CRMs)*, where each CRM consists of a few hundred base pairs, and contains multiple binding sites for multiple transcription factors (TFs) [22, 9, 7]. Methods for identifying CRMs and their component TFs can thus reveal the organization of the transcriptional network in the cell.

In principle, one could use a two-phase approach for identifying CRMs in a set of upstream regions of co-regulated genes. The first phase would scan for single motifs that are enriched in the upstream regions (see, e.g., [2, 13]). The second phase would then try to find correlations between these enriched motifs. Such an approach is suitable for discovering some types of CRMs, like the one depicted in Figure 1(a). However, since each motif is considered in isolation, this approach will fail to discover more subtle CRMs, in which no single motif is enriched, as exemplified in Figure 1(b). CRMs of the latter type can be found by approaches that look for combinations of motifs that exhibit functional synergism, or tend to co-occur in sequences of interest [20, 12, 19, 15]. However, since these methods do not constrain the occurrences of motifs in each combination to be close together within the upstream region, they will fail to discover CRMs of the type shown in Figure 1(c). Recently, several methods have been suggested to identify occurrences of known CRMs [3, 5] and to find novel CRMs of known motifs [16], but these methods do not identify novel motifs and require an annotated list of binding sites.

In this paper we propose a novel model for transcriptional regulation, based on probabilistic graphical models [10], and an algorithm to learn this model automatically from data. Our input consists of a set of putatively co-regulated genes and their raw sequence data. The model has three components. The first is a motif model that describes the probability that a gene contains a binding site for some motif given the upstream region sequence of the gene. In the second component, we consider sequence windows of a prescribed length along the gene's upstream region. For each

window, we model the probability that it contains a CRM that involves k specific motifs, given the binding site occurrences of these motifs. The third component models the probability that a gene contains a CRM given the CRM occurrences in each of the considered windows. We propose an iterative algorithm, based on the expectation maximization (EM) algorithm, for learning the model parameters, and a cross-validation procedure to test the significance of the learned CRMs. Our unified framework generalizes existing approaches for finding CRMs, by integrating both a model for TF binding sites and a model for their organization into modular units. In particular, our method learns motifs de-novo and is suitable for identifying all types of CRMs depicted in Figure 1.

A key property of our model is that it is *discriminative* [15, 17]: Given a set of upstream regions of co-regulated genes, and a background set of upstream regions, the model only attempts to find combinations of motifs that *discriminate* between the two sets. This is in contrast to the common *generative* approaches, that try to build a model of the upstream region sequences, and train the parameters such that the model gives the given sequences a high probability. These approaches can often be confused by repetitive motifs that occur in many upstream regions. These motifs have to be filtered out by using an appropriate background distribution [18]. As we show, our discriminative model allows us to avoid the problem of learning these background distributions, and focus on the classification task at hand.

We evaluated the performance of our method on simulated and real data. On simulated data, our method outperformed extant approaches, and recovered planted CRMs with high accuracy. On real yeast data, we identified significant CRMs in 12 out of 25 tested gene sets that are putatively regulated by two cooperating TFs. In the majority of cases in which the motifs for the corresponding TFs were known (7 out of 11), our method recovered them correctly. Finally, we applied our method to a large collection of human gene sets, derived from the GO process categorization [1]. Overall, we identified 83 significant CRMs, that spanned a diverse set of functional annotations. Many of these CRMs consisted of motifs that were known in the literature, providing additional support that our learned CRMs indeed correspond to true cis-regulatory signals in human.

2. THE PROBABILISTIC MODEL

In this section we present our model of cis-regulation. We model a CRM that consists of k distinct binding site motifs for k TFs, in the upstream region sequences of a set of genes $\mathbf{G}$, where each gene is either regulated by the CRM or not. Thus, we associate a binary *Regulation* attribute R and an upstream region sequence attribute S with each gene. Since we expect a CRM to span a relatively short region, we partition the upstream region S into n shorter overlapping sequence *windows*, where each window has length L. The model then considers CRM occurrences only within these windows.

Our model has three components. The first is a *motif model*, which represents the motif binding sites that are bound by each of the k TFs. We use the motif model to define n binary attributes for each TF i, $g.M_{i1} \ldots g.M_{in}$, indicating whether each of the n windows contains a binding site for the TF. The second component is a *module model*, which represents a CRM as a combination of individual motifs. We use the module model to define n binary attributes, $g.W_1, \ldots g.W_n$, corresponding to whether the CRM appears in each of the n sequence windows. The last component is a *regulation model*, that models the regulation of a gene, $g.R$, by the CRM, as a function of the CRM occurrences in the n different windows. The full model is shown in Figure 2. In the following we describe each of the model components in detail.

2.1 Motif Model

The first component in our model is a set of variables that represent the binding site motifs for each of k transcription factors. For each gene g, we have a set of binary-valued *Motif* variables, $\mathbf{M} = \{g.M_{11} \ldots g.M_{kn}\}$, where $g.M_{ij}$ takes the value *true* iff motif i appears in the j-th sequence window of g. Thus, we allow the motif to play a regulatory role in controlling the expression of gene g, by being a part of the CRM in some windows. We model each motif using the standard *position specific scoring matrix* (PSSM) representation [2, 13], which assumes independence between positions in a binding site. This model assigns a weight to each position in the motif and each nucleotide $\ell \in \{A, C, G, T\}$, representing the extent to which the nucleotide's presence in this position is associated with the motif.

When learning PSSMs, our goal is to estimate the probability that a transcription factor binds a certain gene given its upstream region. Hence, we adapt the discriminative motif model of Segal *et al.* [14], which is well suited for this purpose. This model is specified using a logistic function with p position-specific weights $w_i[\ell]$, one for each position i and each letter $\ell \in \{A, C, G, T\}$, and a threshold w_0. For a window sequence of length L, we assume that binding occurs once, and with equal probability at each of the $L-p+1$ possible positions in the sequence. The probability of binding given the sequence is then specified as:

$$P(g.M = \mathit{true} \mid g.S_1, \ldots, g.S_L) = \mathit{logit}\left(\log\left(\frac{w_0}{L-p+1}\sum_{j=1}^{L-p+1}\exp\{\sum_{i=1}^{p} w_i[g.S_{i+j-1}]\}\right)\right),$$

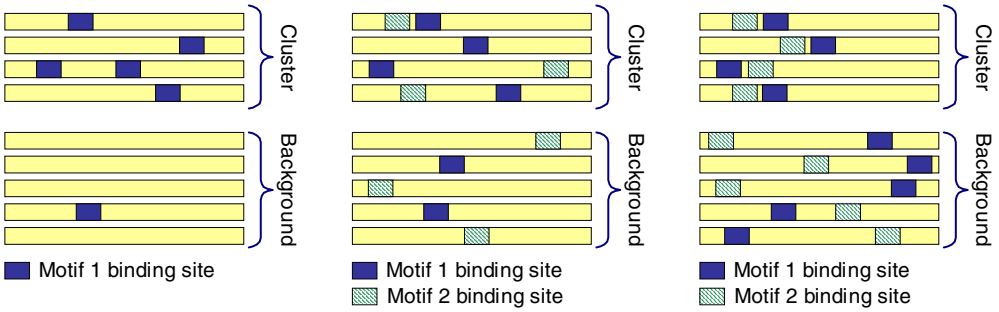
where $\mathit{logit}(x) = \frac{1}{1+e^{-x}}$ is the logistic function. We refer the reader to [14] for additional details.

2.2 Module Model

The second component in our model describes the composition of a CRM in terms of its component motifs. To capture the notion that some motifs may be more important for a particular CRM than others, we model a CRM as a weighted combination of individual motifs. Specifically, we use the logistic function for representing the probability that a sequence window contains the CRM, given the occurrences of the individual motifs in the sequence:

$$P(g.W_j = \mathit{true} \mid g.M_{1j}, \ldots, g.M_{kj}) = \mathit{logit}(v_0 + \sum_{i=1}^{k} v_i \cdot g.M_{ij})$$

where $g.W_j$ is a binary variable representing whether the j-th sequence window contains the CRM, $g.M_{ij}$ is a binary variable representing whether the motif bound by transcription factor i is present in the j-th window, and v_i is a weight that specifies the extent to which motif i plays a regulatory role in the CRM. As the probability that a window contains



Model type	Success (a)	(b)	(c)
Single motif	+	-	-
Motif combinations	+	+	-
Spatial motif combinations	+	+	+

(d)

Figure 1: Comparison of the ability of different methods to detect different types of CRMs. In all cases, shown are the gene upstream regions and the locations of binding sites within them, where genes in the "Cluster" contain the CRM, and genes in the background do not. (a) CRM consisting of a single motif. (b) CRM consisting of a combination of two motifs. (c) CRM consisting of a combination of two motifs that are spatially close to each other. (d) Methods that search for a single motif can only find CRMs of type (a). Methods that search for motif combinations but disregard their spatial relationships cannot find CRMs of type (c). Our proposed method can find CRMs of all types shown.

the CRM depends on $\sum_{i=1}^{k} v_i \cdot g.M_{ij}$, the higher v_i is, the more it contributes to this probability. For interpretability considerations, we restrict the motif weights to be positive (except for v_0). Intuitively, this means that a CRM can only depend on the presence of certain motifs and not on the absence of motifs. We use the CRM model to define n binary window variables for each gene, $g.W_1, \ldots g.W_n$, where the variable for the j-th window, $g.W_j$, depends on the motif occurrences in the j-th window, $g.M_{1j}, \ldots, g.M_{kj}$. Note that the same logistic model is shared across all genes and all windows.

2.3 Regulation Model

The last component in our model combines the information from each window to specify whether the gene is indeed regulated by the CRM. This model follows our intuition that the probability that a gene is regulated by a CRM increases with the number of windows in its upstream region that contain the CRM. The model describes this regulation probability using a logistic function:

$$P(g.R = true \mid g.W_1, \ldots, g.W_n) = logit(p_0 + \sum_{i=1}^{k} p_i \cdot g.M_i),$$

where $g.W_i$ corresponds to whether window i contains the CRM, and p_i specifies the extent to which the presence of the CRM in window i contributes to the overall probability that the gene is regulated. If we expect a priori that certain sequence windows are more likely to contain the CRM than others, then we can assign a higher weight to those windows. For example, when searching for CRMs in human, we might assign a higher weight to those sequence windows that are more conserved between human and mouse. In our setting, we assume that all windows are equally likely to contain the CRM and, thus, use the same weight for all windows. As we show later, this assumption leads to significant computational advantages.

2.4 Unified Model

We combine the above three components into a unified probabilistic graphical model, shown in Figure 2. The model defines the following joint distribution:

$$P(g.R, g.\mathbf{W}, g.\mathbf{M} \mid g.S) = P(g.R \mid g.\mathbf{W}) \cdot \prod_{j=1}^{n} \left(P(g.W_j \mid g.M_{1j}, \ldots, g.M_{kj}) \prod_{i=1}^{k} P(g.M_{ij} \mid g.S_j) \right), \quad (1)$$

where $g.S_j$ is the sequence of window j, and each of the above conditional probability distributions is parameterized as described in the previous sections. Given a model parameterization, we can compute the probability that a gene is regulated by the CRM given the sequence:

$$\begin{aligned} &P(g.R = true \mid g.S) \\ &= \sum_{\bar{w} \in \mathbf{W}} P(g.R = true \mid g.\mathbf{W} = \bar{w}) \\ &\quad \cdot \sum_{\bar{m} \in \mathbf{M}} P(g.\mathbf{W} = \bar{w} \mid g.\mathbf{M} = \bar{m}) P(g.\mathbf{M} = \bar{m} \mid g.S) \\ &= \sum_{\bar{w} \in \mathbf{W}} P(g.R = true \mid g.\mathbf{W} = \bar{w}) \\ &\quad \cdot \prod_{j=1}^{n} \sum_{\bar{m} \in \mathbf{M}[j]} P(g.W_j = \bar{w}[j] \mid g.\mathbf{M}[j] = \bar{m}) \\ &\quad \cdot \prod_{i=1}^{k} P(g.M_{ij} = \bar{m}[i] \mid g.S) \end{aligned}$$

where $\bar{w}$ is a vector that ranges over all possible assignments to each of the n window variables, $\bar{m}$ is a vector that ranges over all possible assignments to each of the $k \cdot n$ motif variables, and $\mathbf{M}[j]$ corresponds to the set of motif variables for window j, $M_{1j}, \ldots, M_{kj}$.

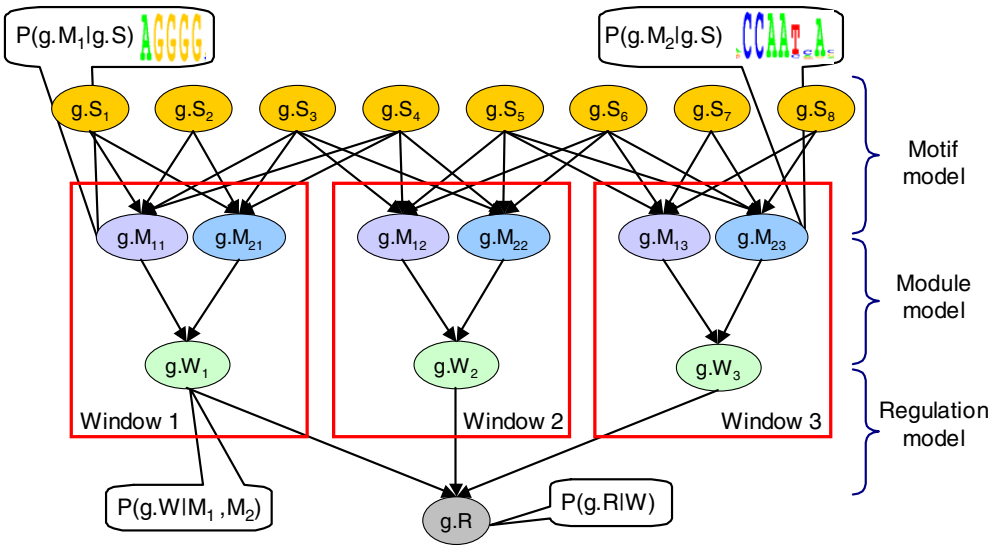


Figure 2: Illustration of our unified model, for a simple example with upstream regions of length eight, windows of length four with two base-pair overlaps, and two motifs. The model contains a total of four distinct CPDs (shown). The CPDs for the first motif are the same and hence $P(g.M_1 \mid S) = P(g.M_{11} \mid S) = P(g.M_{12} \mid S) = P(g.M_{13} \mid S)$. Similarly, the CPDs for the second motif are the same and hence $P(g.M_2 \mid S) = P(g.M_{21} \mid S) = P(g.M_{22} \mid S) = P(g.M_{23} \mid S)$. Finally, the same CPD is shared across all windows and hence $P(W \mid M_1, M_2) = P(W_1 \mid M_1, M_2) = P(W_2 \mid M_1, M_2) = P(W_3 \mid M_1, M_2)$.

3. LEARNING THE MODEL

In the previous section we presented our probabilistic model. We now turn to the task of learning this model from data. Our training data set D consists of a set of genes $\mathbf{G}$, where for each gene g we are given its upstream region sequence $g.S$ and the value of $g.R$, indicating whether g is regulated by the CRM or not. In learning the models, we need to estimate the model parameters, which include: the weights of the k PSSMs; the weights of the logistic distribution $v_0, \ldots, v_k$ for the module model $P(g.W \mid g.\mathbf{M})$; and the weights of the logistic distribution $p_0, \ldots, p_n$ for the regulation model $P(g.R \mid g.\mathbf{W})$.

We follow the standard approach of *maximum likelihood* estimation: Find the parameters $\boldsymbol{\theta}$ that maximize $P(D \mid \boldsymbol{\theta})$. Our learning task is made considerably more difficult by the fact that both the *Motif* variables $g.\mathbf{M}$ and the *Window* variables $g.\mathbf{W}$ are unobserved in the training data. In this case, the likelihood function has multiple local maxima, and no general method exists for finding the global maximum. We thus use the *Expectation Maximization (EM)* algorithm [4], which provides an approach for finding a local maximum of the likelihood function. Starting from an initial guess $\boldsymbol{\theta}^{(0)}$ for the parameters, EM iterates the following two steps. The *E-step* computes the distribution over the unobserved variables given the observed data and the current estimate of the parameters. The *M-step* then re-estimates the parameters by maximizing the likelihood with respect to the distribution computed in the E-step. This estimation task differs for the different parts of the model.

3.1 E-step: Inferring Modules and Regulation

Our task in the E-step is to compute the distribution over the unobserved data, which in our setting means computing $P(g.\mathbf{W}, g.\mathbf{M} \mid g.S, g.R)$. As genes are assumed to be independent, this computation can be done separately for each gene, by performing inference in the Bayesian network of Figure 2. Moreover, since the sequences variables $g.S$ are always observed, the network in which we need to perform inference is effectively a tree. Hence, inference can be performed efficiently using the *clique tree* algorithm [10].

In general, the computations carried out by the clique tree algorithm are exponential in the number of parents of each node in the network. In our case, this means that the E-step will be exponential in the number of *Motif* and *Window* variables. As the number of motifs k in a CRM is typically small ($k \leq 5$), our main computational concern is with the number of windows. In a typical setting, we might search for CRMs in upstream regions of length 1000bp, using windows of length 200bp with an overlap of 100bp between windows. In this case, we have 9 windows and the E-step can be computed efficiently. However, there might be settings in which we wish to search for CRMs in longer upstream regions, or using more overlap between windows. In these settings, exact inference is infeasible.

When the number of windows is prohibitively large, we propose to use the *hard assignment* version of the EM algorithm. In this version, the E-step computes the most likely assignment to the hidden variables, and the M-step then re-estimates the parameters by maximizing the likelihood with respect to the assignment computed in the E-step. Under the assumption that all sequence windows are equally likely to contain the CRM, it turns out that we can find the most likely assignment to the hidden variables in time that is linear in the number of windows. The algorithm is based on the observation that if the weights p_i in the logistic function $P(g.R \mid g.\mathbf{W})$ are the same for all windows, then the value of the first term in Equation 1 is a function of the number t of window variables whose assignment is *true*, and does not

depend on which window variables are actually set to *true*. Hence, under the constraint that exactly t of the window variables are assigned to *true*, the problem of finding the most likely assignment can be reduced to finding:

$$\begin{aligned}\{\bar{w},\bar{m}\} &= \text{argmax}_{\bar{w}',\bar{m}'} P(g.\mathbf{W}=\bar{w}' \mid g.\mathbf{M}=\bar{m}') \\ &\quad \cdot P(g.\mathbf{M}=\bar{m}' \mid g.S) \\ &= \text{argmax}_{\bar{w}',\bar{m}'} \prod_{j=1}^{n} P(g.W_j=\bar{w}'[j] \mid g.\mathbf{M}[j]=\bar{m}'[j]) \\ &\quad \cdot P(g.\mathbf{M}[j]=\bar{m}'[j] \mid g.S)\end{aligned}$$

where $\bar{w}$ and $\bar{m}$ range over all possible assignments to the *Window* and *Motif* variables, respectively. Thus, the computation decomposes by windows, and the most likely assignment under this constraint is to assign to *true* the t window variables with the highest value of $\max_{m'} P(g.W = \mathit{true} \mid g.\mathbf{M}=m')P(g.\mathbf{M}=m' \mid g.S)/P(g.W=\mathit{false} \mid g.\mathbf{M}=m')P(g.\mathbf{M}=m' \mid g.S)$. Finally, we choose t as the integer $0 \le t \le n$, which yields the assignment with the highest probability.

3.2 M-step: Estimating Model Parameters

In the M-step, our goal is to estimate the parameters for the distribution of each component of the model so as to maximize the conditional log probability of that component. For the motif model, this means estimating the parameters $P(g.M_i \mid g.S)$ for each motif i of the k motifs, that maximize $\sum_{g\in\mathbf{G}}\sum_{j=1}^{n}\sum_{m\in Mij} E[M_{ij}=m]\log P(g.M_{ij}=m \mid g.S)$, where m ranges over the possible assignments to M_{ij}, $\{\mathit{false},\mathit{true}\}$, and $E[M_{ij}=m]$ is computed in the E-step and is equal to the probability $P(g.M_{ij}=m \mid g.S,g.R)$. Unfortunately, this optimization problem has no closed form solution, and there are many local maxima. We therefore use a conjugate gradient ascent to find a local optimum in the parameter space.

For the module model, we need to estimate the logistic weights of each motif, $w_0,\ldots,w_k$, in the distribution $P(g.W \mid g.\mathbf{M})$ that maximize $\sum_{g\in\mathbf{G}}\sum_{j=1}^{n}\sum_{m\in\mathbf{M}_j}\sum_{w\in W_j} E[W_j=w,\mathbf{M}_j=m]\log P(g.W_j=w \mid g.\mathbf{M}_j=m)$, where $\mathbf{M}_j=\{M_{1j},\ldots,M_{kj}\}$, m and w range over the possible assignments to $\mathbf{M}_j$ and W_j, respectively, and $E[W_j=w,\mathbf{M}_j=m]$ is computed in the E-step and is equal to the probability $P(g.W_j=w,g.\mathbf{M}_j=m \mid g.S,g.R)$. Each weight is also constrained to be positive (see Section 2.2). Although there is no closed for this constrained optimization problem, the target function is convex, allowing us to find the optimal parameter estimates using gradient ascent on the target function.

Finally, we need to estimate the window weight parameters of the distribution $P(g.R \mid g.\mathbf{W})$ that maximize $\sum_{g\in\mathbf{G}}\sum_{w\in\mathbf{W}} E[R=r,\mathbf{W}=w]\log P(g.R=r \mid g.\mathbf{W}=w)$, where w ranges over the possible assignments to $\mathbf{W}$, r indicates whether g is regulated, and $E[R=r,\mathbf{W}=w]$ is computed in the E-step and is equal to the probability $P(g.R=r,g.\mathbf{W}=w \mid g.S)$. This is a similar optimization problem as in the module model case, and we thus apply gradient ascent to find the optimal parameter setting.

3.3 Model Initialization

In the previous sections we showed how to apply the EM algorithm to improve the quality of the model in every iteration, and converge to a local maximum of the likelihood function. However, the EM algorithm requires an initial model parameterization, which we need to provide. As for all applications of EM, the quality of the starting point has a large impact on the quality of the local optimum found by the algorithm. This is in particular true for the parameters of the motif model (for each motif).

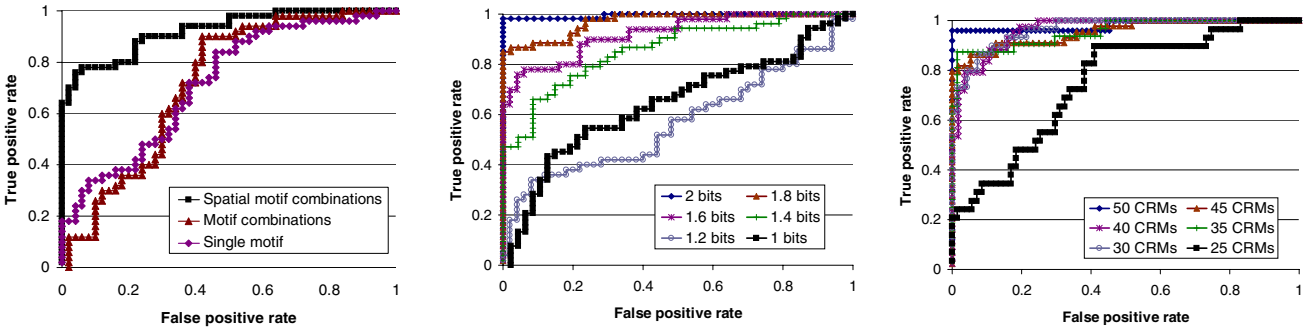
We devised a two-phase scheme for the initialization of the motif parameters. In the first phase, we efficiently generate motif seeds of fixed length (e.g., 6-9), that are abundant in the upstream regions of the regulated genes. We use the identified seeds to initialize motifs by considering occurrences of these seeds with at most one mismatch. These occurrences allow us to initialize a PSSM for each seed and also to possibly extend it in each end by positions whose information content exceeds a threshold. In the second phase, we score combinations of up to k motif seeds, using the hypergeometric significance test, allowing us to find motif combinations that *discriminate* between the regulated genes non-regulated ones. Thus, even in this initialization step, we search for combinations of motifs rather than individual motifs, as this initialization is more suited for the types of CRMs we wish to find.

4. EXPERIMENTAL RESULTS

We applied our module identification method to simulated and real data. The goal in the simulations was to test the ability of the algorithm to recover planted CRMs. In real data we wished to evaluate the performance of the algorithm in recovering known modules in yeast, and to apply it to discover novel modules in human. In all cases, the only input to our program was a set of upstream regions, the window length L, and the list of regulated genes for which $g.R = \mathit{true}$, whose upstream regions is expected to contain the CRM. We designed all models such that a gene is regulated even if only one of its sequence windows contains the CRM, by fixing the weights, p_i, of all windows in the regulation model of Section 2.3 to 12, and setting $p_0 = -6$. By fixing these weights, the learning algorithm tries to find CRMs that do not occur in windows of background genes and occur at least once in the sequence windows of the regulated genes. While this results in more interpretable models, it brings up a practical consideration, which is that most of the sequence window variables, and consequently most of the motif variables, will be set to *false*, leading to an unbalanced optimization problem when updating the weights of each motif. Thus, in practice, we balance this optimization problem by only considering the window with the highest posterior for each gene.

4.1 Simulated Data

As a basic test of our procedure in a controlled setting, we generated random upstream region sequences of length 400 for 50 regulated and 50 non-regulated genes, and planted CRMs consisting of two motifs of length 8 in a varying fraction of the regulated genes. This gives a known ground truth to which we can compare the learned models. To make the data realistic, we also planted both motifs in 25 of the non-regulated genes, but unlike the motif occurrences in the regulated genes, that were constrained to appear in proximity within the upstream regions, we randomly distributed the two motifs of these 25 non-regulated genes within the upstream regions. Our setting is thus designed such that algorithms that search for a single motif, or algorithms that



(a) (b) (c)

Figure 3: Performance on simulated data, shown as ROC curves, where the x-axis is the *false positive rate*, $FP/(FP+TN)$, and the y-axis is the *true positive rate*, $TP/(TP+FN)$. In all cases, both motifs were planted in 25 of the non-regulated genes. (a) Comparison of different methods when the motifs were planted in all 50 regulated genes and sampled from PSSMs with 1.5 bits of information per each of the 8 positions. (b) Performance as a function of binding specificity. In each dataset the motifs were planted in all 50 regulated genes, but were sampled from motifs with varying bits of information per position. (c) Performance as a function of the fraction of regulated genes in which the motifs were planted, where the planted motifs were sampled from a PSSM with 1.5 information bits per position.

search for motif combinations but ignore the spatial locations of motifs, will not succeed. Indeed, our algorithm recovered the planted motifs with high accuracy, whereas the above methods did not, as shown in the comparison of the ROC curves of Figure 3(a). These curves compare the *false positive rate* to the *true positive rate*, when changing the probability threshold at which the *Regulates* variable, $g.R$, is considered to contain the CRM. As transcription factors vary greatly in their binding specificity, it is important that our method can recover CRMs whose motifs exhibit variation in their actual instances. To test this ability, we generated six different data sets, where in each case we varied the information per bit in the PSSM from which we sampled the planted motifs. The results of applying our method to each of these datasets are shown in Figure 3(b), indicating that most of the planted motifs are recovered even when there is large variation in their instances. The input to our method includes a set of co-regulated genes that are expected to share a CRM. As this input set may contain errors, it is important that we recover CRMs even when only a fraction of the input regulated genes contain it. To test this, we applied our method to six different datasets that varied in the fraction of regulated genes in which we planted the CRM. Our results, in Figure 3(c), show very good performance even when the CRM was planted in only 30 of the 50 co-regulated genes, slightly more than the 25 confounding occurrences of the motifs in the non-regulated genes.

4.2 Cis-Regulatory Modules in Yeast

To evaluate the performance of our method in a more realistic setting, we tested its ability to detect putative cis-regulatory modules in yeast. As the collection of CRMs in the literature is limited, we used the genome-wide location data of Lee *et al.* [8] to compile a collection of gene sets for which strong experimental evidence suggests that the genes in each set are regulated by the same two transcription factors. We hypothesized that the genes in each such set should thus contain a CRM consisting of the binding sites for the two TFs. Specifically, the location data contains genome-wide Chromatin-Immunoprecipitation experiments for 106 yeast TFs, where each experiment measured the relative occupancy of the upstream regions of all yeast genes by the TF. We considered measurements with $p < 0.001$ as indicating that the TF binds the upstream region of the corresponding gene. Thus, with each TF we associated a set of target genes to which the TF binds in-vivo. To obtain gene sets that are regulated by two TFs, we computed the intersection of the targets of every pair of TFs, and kept only those intersections with at least 25 genes, such that the size of the intersection was greater than would be expected by chance (scored using a hypergeometric distribution). Altogether, we found 25 such gene sets. We hypothesized that each such set contains a CRM corresponding to the two TFs, and applied our method to each set using 100bp and 200bp windows, and 500bp upstream regions for each gene. In each case, we took the genes in the intersection set to be regulated ($g.R = true$), and selected 100 random genes for which we assumed regulation does not take place ($g.R = false$).

To evaluate the quality of the CRMs we learned, we tested whether they captured some characteristics that are specific to the regulated genes. To this end, we performed leave-one-out experiments, where in each experiment we learned a CRM using all the genes except for one, and then used the learned CRM model to compute the probability that the held out gene is regulated by the CRM. If the CRM is indeed specific to the regulated genes, then regulated genes that are held out should receive a higher probability for being regulated than the held out genes that were selected at random. We measured this by computing the *classification margin*: The largest difference between the fraction of held out regulated genes whose regulation probability is above some threshold t, and the fraction of held out non-regulated genes whose regulation probability is above t, for different values of t. To evaluate the significance of the margins we obtained, we compared them to those obtained on 100 datasets, in which random yeast genes were assigned random labels (50 regulated and 50 not regulated).

We detected significant CRMs in 12 out of our 25 sets

TF pair	Margin	Known motifs	Predicted motifs	%Correct
FHL1, YAP5	0.688	-	ATGTAAGG, CCGTACAT	-
FHL1, RAP1	0.687	ACACCCATACATTT (RAP1)	AATGTATG, CCATACAT (RAP1)	1/1
SWI4, SWI6	0.592	TTTTCGCG (SWI4), ACGCGT (SWI6)	TTTTCCCG (SWI4), AACGCGAA (SWI6)	2/2
MBP1, SWI6	0.538	ACGCGTnA (MBP1), ACGCGT (SWI6)	GACGCGTA (MBP1), CGACGCGA (SWI6)	2/2
ACE2, SWI5	0.518	GCTGGT (ACE2), KCGTGR (SWI5)	ACACACACACA	0/2
GAT3, RGM1	0.484	-	GGGTGTGT, CGCCCCCCA	-
FKH2, MCM1	0.46	TTGTTTACST (FKH2) TTWCCCnWWWRGGAAA (MCM1)	CCCTTTTC (MCM1), AGTAAACA	1/2
RAP1, YAP5	0.448	-	ATTTATGG, TCCATCAC	-
NDD1, SWI4	0.447	TTTTCGCG (SWI4)	TGTGCGTG, CACTCACAC	0/1
GAT3, YAP5	0.428	-	CTCAACTA, CTATCTGA	-
GAT3, PDR1	0.423	CCGCGG (PDR1)	AAGCGGCTGA (PDR1), TCGTTGCTC	1/1
NRG1, YAP6	0.414	-	ATACGAAA, GATAGGCA	-

Table 1: Significant CRMs discovered in yeast ($p < 0.01$). For each module, shown are the two TFs that putatively regulate the genes in each input set; their consensus; the consensus of the learned motifs; and the correspondence between the known motifs and learned ones. We considered two consensus sequences as matching, if one was a subsequence of the other with at most two mismatches.

($p < 0.01$). These CRMs are summarized in Table 4.1. Since each input gene set is the intersection of the targets of two TFs, we expect the CRM to consist of the binding sites for the corresponding TFs. Thus, we further validated our learned CRMs by comparing the consensus sequence of their motifs to their consensus according to Kellis *et al.* [6]. Our learned motifs matched the known ones very well, recovering 7 out of the 11 known motifs. For 13 sets we did not discover a significant CRM. This may be explained by the small size of the gene sets (most sizes ranged between 25 and 30) and by the fact that multi-factorial regulation does not necessarily involve modular structures. Overall, the results on the location dataset demonstrate the ability of our method to detect true signals in real data.

4.3 Cis-Regulatory Modules in Human

Discovering cis-regulatory information in human is hard compared to yeast, as genes are typically regulated by a combination of several TFs and the sequence regions involved in the regulation are often farther from the transcription start site. We tested whether our method, which is designed for discovering these more complex regulatory signatures, can detect true CRMs in human. As the input gene sets, we used sets of genes that are known to be involved in the same process according to the GO database [1]. We hypothesized that such sets are likely to be regulated by several TFs and thus their upstream regions might contain a CRM. Specifically, we extracted all GO annotations with at least 25 genes, but less than 150 genes, and applied our method to each of the 381 such annotations, using 200bp windows with 100bp overlap between windows, and 1000bp upstream region for each gene[1]. For each GO process category, we treated its member genes as regulated by a common CRM ($g.R = true$), and selected 100 random genes to serve as a negative set ($g.R = false$).

As few CRMs are known in human, we evaluated the quality of the CRMs that we learned using the leave-one-out procedure described above. For each CRM, we measured the classification margin of its leave-one-out experiment, and compared it to the classification margin obtained on 100 sets of random human genes. Overall, we found 83 significant CRMs, spanning 71 GO categories ($p < 0.01$), where 46 of these CRMs consisted of two motifs, and 37 consisted of three motifs, for a total of 203 motif instances. We matched this list of motifs against a list of 414 known motifs from Wingender *et al.* [21], using the comparison method of Pietrokovski *et al.* [11]. Out of the 203 motif instances that we learned, 54 corresponded to known motifs, spanning 36 distinct motifs. The leave-one-experiments, combined with the recovery of known motifs, provide strong evidence that our method indeed detected a large number of putatively true CRMs in human. A summary of all of the significant CRMs that we found, including the GO category that was used as input and the known motifs that were recovered, is shown in Figure 4.

A more detailed inspection of our results showed many GO classes for which at least one of the motifs that we learned was known in the literature to be bound by a TF that regulates the genes associated with that class. For example, we learned a significant CRM for protein folding genes, in which one of the motifs was the binding site for HSF (Heat Shock Factor), a known activator of protein folding genes under stress and heat shock conditions. As another example, one of the motifs we learned for the CRM of mitochondrial membrane genes was the binding site for the GATA TF, which is known to induce mitochondrial membrane genes. We also inspected the learned CRMs visually, and found that they indeed consisted of motifs whose occurrences were close to each other in the upstream region of the regulated genes, whereas these motifs did not occur very often in the non-regulated genes. An example is shown in Figure 5 for the CRM learned from the "regulation of CDK activity" class. As can be seen, for this category, 13 of the 28 genes contain the CRM. In contrast, this CRM appears in only 4 of the 100 non-regulated genes (data not shown). As further support for this CRM, one of the motifs composing this CRM was the binding site for NKX, a regulator of insulin biosynthesis, which also has some known role in regulating cyclin dependent kinase (CDK) genes.

5. DISCUSSION

In this paper we presented a novel model of the mechanism of cis-regulation, which captures many aspects of this process, including the presence of multiple binding sites for multiple transcription factors in short DNA sequences. We

[1]We experimented several parameter settings. The choice of 200bp windows with 100bp overlap gave the best results, though other similar settings yielded similar results.

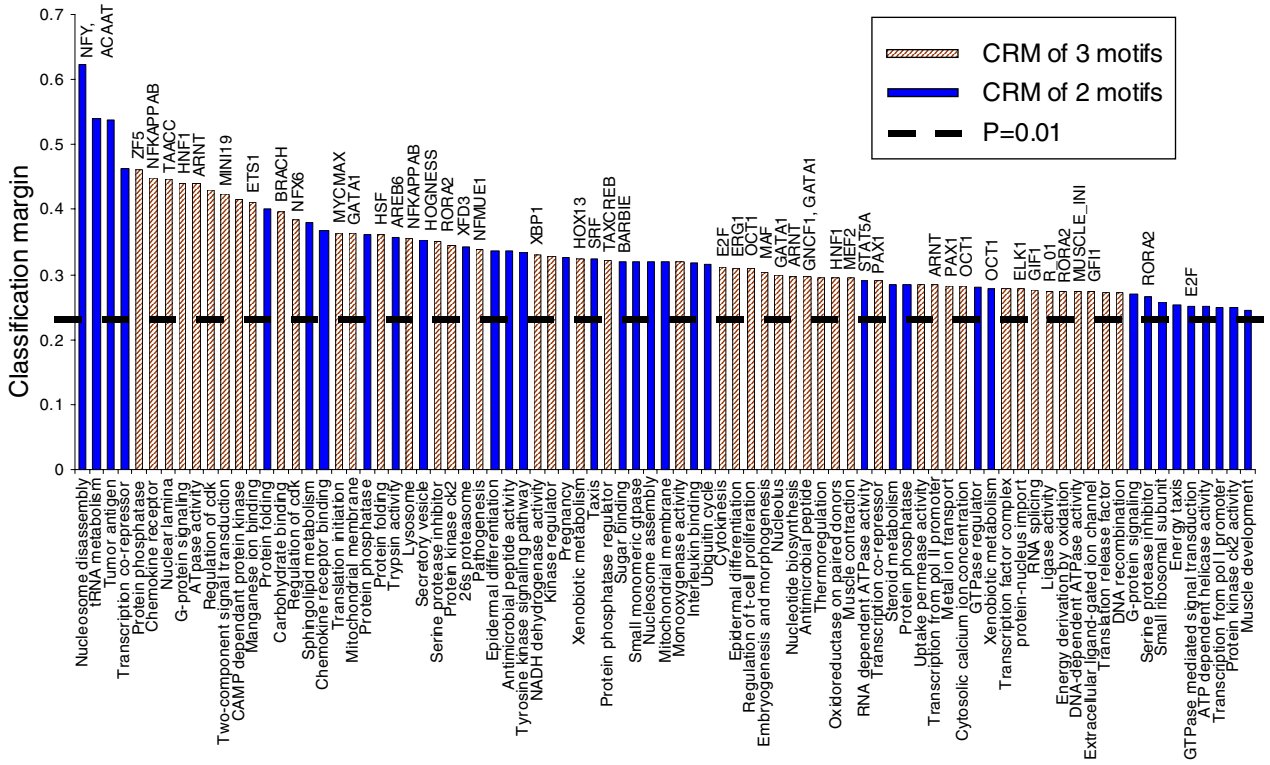


Figure 4: Summary of all 83 significant CRMs ($p < 0.01$) that we learned in human, sorted by the classification margin (y-axis) obtained for each CRM in leave-one-out experiments. The x-axis indicates the GO category that was used as the input gene set when learning the CRM. In cases where one or more of the motifs that we learned for the CRM was known, we listed it above the classification margin of the corresponding CRM. The dashed black line indicates the best classification margin obtained from applying our method to 100 sets of random human genes, and thus corresponds to $p = 0.01$.

presented an algorithm to learn this model from data, which allows us to predict cis-regulatory modules and their component motifs using only the raw sequence data as input. Our results demonstrated the ability of our method to find known signals in simulated data and in yeast, and showed its utility for detecting an extensive list of significant modules in human.

There are several directions for refining and extending our approach. First, our model requires a specification of the sequence windows in which we expect to find the CRM. We are now working on modifications to the model that will treat the entire upstream region as one sequence, but still bias the search towards finding motifs whose occurrences are next to each other. Second, we are exploring the use of our approach as part of a richer probabilistic framework that combines gene expression measurements [15]. Finally, in some cases we did not detect significant CRMs. While some of these may be due to limitations of our approach, understanding the reasons for failing in the other cases may reveal novel characteristics of cis-regulation.

Acknowledgments

ES was supported by a Stanford Graduate Fellowship (SGF). RS was supported by a Fulbright grant. This research was also supported in part by NSF ITR Grant CCR-0121555.

6. REFERENCES

[1] M. Ashburner and *et al.*. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25:25–9, 2000.

[2] 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.

[3] B. P. Berman, Y. Nibu, B. D. Pfeiffer, P. Tomancak, S. E. Celniker, M. Levine, G. Rubin, and M. B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the drosophila genome. *PNAS*, 2:757–62, 2002.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–39, 1977.

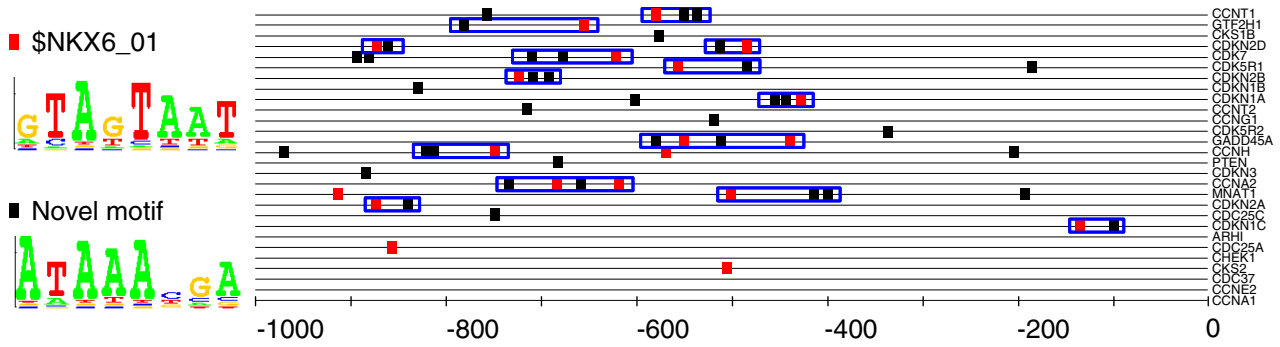


Figure 5: Visualization of the two motif CRM that we learned for the regulation of CDK activity annotation. Shown are the 1000bp upstream regions of all 28 genes that are assigned to this annotation according to GO. The occurrences of the two motifs in each upstream region is shown as red (NKX motif) and black (novel motif) rectangles, where we highlighted (blue rectangle) all 13 genes in which the two motifs occurred within 200bp of each other in the gene's upstream region.

[5] M.C. Frith, U. Hansen, and Z. Weng. Detection of cis-element clusters in higher eukaryotic DNA. *Bioinformatics*, 17(10):878–889, 2001.

[6] M. Kellis et al. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423:241–254, 2003.

[7] W. Krivan and W.W. Wasserman. A predictive model for regulatory sequences directing liver-specific transcription. *Genome Res.*, 11(9):1559–66, 2001.

[8] T.I. Lee, N.J. Rinaldi, F. Robert, et al. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.

[9] M.Z. Ludwig, N.H. Patel, and M. Kreitman. Functional analysis of eve stripe 2 enhancer evolution in drosophila: rules governing conservation and change. *Development*, 125(5):949–58, 1998.

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

[11] S. Pietrokovski. Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nuc. Acids Res.*, 19:3836–45, 1996.

[12] Y. Pilpel, P. Sudarsanam, and G.M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat. Genet.*, 29(2):153–9, 2001.

[13] 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.

[14] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From sequence to expression: A probabilistic framework. In *Proc. RECOMB*, 2002.

[15] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, 19(Suppl 1):S273–82, 2003.

[16] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. Karp. Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, 19(Suppl 1):S283–91, 2003.

[17] S. Sinha. Discriminative motifs. In *Proc. RECOMB*, pages 291–298, 2002.

[18] 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.

[19] D.G. Thakurta and G.D. Stormo. Identifying target sites for cooperatively binding factors. *Bioinformatics*, 17(7):608–621, 2001.

[20] W.W. Wasserman and J.W. Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.*, 278(1):167–81, 1998.

[21] E. Wingender and *et al.*. The TRANSFAC system on gene expression regulation. *Nuc. Acids Res.*, 29:281–283, 2001.

[22] C.H. Yuh, H. Bolouri, and E.H. Davidson. Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902, 1998.