Combining Sequence and Time Series Expression Data to Learn Transcriptional Modules

Anshul Kundaje, Manuel Middendorf, Feng Gao, Chris Wiggins, and Christina Leslie

Abstract—Our goal is to cluster genes into *transcriptional modules*—sets of genes where similarity in expression is explained by common regulatory mechanisms at the transcriptional level. We want to learn modules from both time series gene expression data and genome-wide motif data that are now readily available for organisms such as *S. cereviseae* as a result of prior computational studies or experimental results. We present a generative probabilistic model for combining regulatory sequence and time series expression data to cluster genes into coherent *transcriptional modules*. Starting with a set of motifs representing known or putative regulatory elements (transcription factor binding sites) and the counts of occurrences of these motifs in each gene's promoter region, together with a time series expression profile for each gene, the learning algorithm uses expectation maximization to learn module assignments based on both types of data. We also present a technique based on the Jensen-Shannon entropy contributions of motifs in the learned model for associating the most significant motifs to each module. Thus, the algorithm gives a global approach for associating sets of regulatory elements to "modules" of genes with similar time series expression profiles. The model for expression data exploits our prior belief of smooth dependence on time by using statistical splines and is suitable for typical time course data sets with relatively few experiments. Moreover, the model is sufficiently interpretable that we can understand how both sequence data and expression data contribute to the cluster assignments, and how to interpolate between the two data sources. We present experimental results on the yeast cell cycle to validate our method and find that our combined expression and motif clustering algorithm discovers modules with both coherent expression and similar motif patterns, including binding motifs associated to known cell cycle transcription factors.

Index Terms—Gene regulation, clustering, heterogeneous data.

1 Introduction

Since the earliest papers in functional genomics [8], computational biologists have relied on clustering algorithms as their principal tool for analyzing microarray data. While clustering has provided useful insights and remains popular, it can be difficult to interpret clusters in terms of the underlying mechanisms of gene regulation. Although we understand that genes with extremely similar regulatory (promoter) sequences should be expressed similarly, the converse is generally not true. Moreover, the large systemic noise inherent in such high throughput technologies as microarrays [25] make it difficult to distinguish between subtle expression patterns. Thus, genes within a cluster that are statistically similar in expression can be biologically unrelated in regulatory sequence or function.

Recent biology and computational biology literature [13], [19], [21], [22] has explored the paradigm of a *transcriptional module*—a collection of genes under (perhaps combinatorial) control of a set of transcription factors that bind to regulatory elements in the promoter regions for these genes. Under changes of experimental conditions or in the course of a time series, genes in a transcriptional module should undergo similar changes in mRNA expression. In order to learn the membership of transcriptional modules—or to study the related question of finding sets of regulatory elements that co-occur in promoter regions of genes with a common expression pattern—it is natural to combine both regulatory sequence and expression data in a statistical or learning approach. Pilpel et al. [19] study promoter elements in yeast and define a "motif synergy" score for a pair of motifs, based on comparison of the expression coherence of the set of genes that contain both motifs in their promoters versus those that contain either motif alone. Thus, rather than fully describing transcriptional modules, they study "synergistic" relationships between pairs of regulatory elements. They also explore relationships between sets of N motifs by considering expression coherence of each of the 2^N sets of genes corresponding all possible binary signatures (presence or absence) of the N motifs in the promoter region. Ihmels et al. [13] present an algorithm to assign genes to (overlapping) transcriptional modules, but here the notion of module corresponds to an "expression signature" across experimental conditions, and motif data are not used. Segal et al. [22] developed a model for combining promoter sequence data and a large amount of expression data to learn transcriptional modules on a genome-wide level in *S. cerevisiae*. Their algorithm incorporates a motif discovery procedure for refining the set of putative regulatory elements that help explain assignments of genes to modules, and different data sources are

- *A. Kundaje and C. Leslie are with the Department of Computer Science, Columbia University, New York 10027. E-mail: {abk2001, cleslie}@cs.columbia.edu.*
- *M. Middendorf is with the Department of Physics, Columbia University, New York 10027. E-mail: mjm2007@columbia.edu.*
- *F. Gao is with the Department of Biological Sciences, Columbia University, New York 10027. E-mail: fg2037@columbia.edu.*
- *C. Wiggins is with the Department of Applied Mathematics, Columbia University, New York 10027. E-mail: chris.wiggins@columbia.edu.*

Manuscript received 1 Sept. 2004; revised 12 Dec. 2004; accepted 14 Apr. 2005; published online 31 Aug. 2005.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0119-0904.

integrated in a probabilistic relational model. However, this generative probabilistic model is quite complex and presents significant technical and computational challenges; thus, the method requires considerable expertise to implement, use, or understand. Moreover, the model is appropriate only for large expression data sets rather the smaller data sets that are typically produced by the majority of biological labs. More recently, Beer and Tavazoie [6] used a Bayes network approach to try to learn rules based on sets of motifs in the regulatory sequence of a gene that would predict its cluster membership, relative to a fixed initial clustering of genes by gene expression profiles. Also worth noting is a much earlier paper of Holmes and Bruno [11] who presented a joint probabilistic model for sequence and expression data, but used it for the purpose of motif discovery, adopting a Gibbs sampling approach to find overrepresented motifs.

Our goal in this paper is to address the global problem of learning transcriptional modules in a more constrained setting than in Segal et al. [22]: Starting with occurrence data of putative regulatory elements and a time series expression profile for each gene, we present a probabilistic model for combining the regulatory sequence and expression data in order to cluster genes into coherent putative transcriptional modules. Note that we do not perform motif discovery as part of our learning procedure, but rather assume that we have a list of motifs for putative transcription factor binding sites and the count of their occurrences in each promoter sequence. These data can be readily obtained for *S. cerevisiae* by using any standard motif discovery algorithm [1], [12] or by searching a database of regulatory elements such as TRANSFAC [26]. (In particular, our approach is conceptually different from that of Holmes and Bruno [11] in that we are using motif profiles with respect to a known set of motifs as a source of data rather than seeking to extract a few significant motifs, although both that work and ours use a framework of joint probabilistic learning.) Our learning algorithm is based on expectation maximization (EM) for the underlying generative model and does not require large computational resources. Moreover, the model is simple enough that we can understand how both sequence data and expression data contribute to the cluster assignments, and how to interpolate between the two data sources. Finally, we use an information-theoretic technique, based on the Jenson-Shannon (JS) entropy of the motif parameters in the learned model, to associate to each module a set of significant motifs. In this way, we can hope to learn combinations of motifs that are strongly associated to clusters of genes with coherent expression patterns, without limiting ourselves to finding pairs of correlated motifs or needing to investigate all combinations of motif patterns as in Pilpel et al. [19]. We note that our EM algorithm uses explicit closed-form update rules as described in the text, and that a MATLAB implementation of the code is available from our supplementary Website, so that our results can be easily reproduced and the algorithm applied to other data sets. By contrast, Segal et al. [22] use the more complex formalism of probabilistic relational models: While the training procedure is motivated by expectation maximization, there is no closed form solution for the update rules, and special approximation techniques and careful initialization procedures to seed the optimization problem are required at each iteration of the algorithm. There is currently no publicly available implementation of the code, making it difficult to reproduce the results or apply the algorithm to other problems. Similar to Beer and Tavazoie [6], we can learn a mapping from motifs to clusters. Note, however, that Beer and Tavazoie cluster based on expression alone, use these clusters as input to a motif discovery algorithm to find motifs for each cluster, and then search through Bayes networks that use the chosen motifs to predict cluster membership. In the Beer approach, sequence information is not used to determine the initial clusters, so differently regulated genes with similar expression profiles can be incorrectly grouped together, and only motifs that are overrepresented in promoter sequences for the expression-only clusters will be found; however, the Bayes net model allows a richer modeling of interactions between regulatory elements, including positional preference or orientation. We avoid the structure learning problem for Bayesian networks and obtain clusters of genes through joint learning from expression and sequence data rather than first fixing clusters based on expression profiles alone. Richer modeling of interactions of the significant motifs is not incorporated in our algorithm, but can be implemented as a postprocessing step.

This model was motivated by recent machine learning work related to joint models for text and image data in Web pages [4], where there is a similar problem of combining discrete word count data with more complex continuous data. Our particular interest was to learn modules from time series expression data, since usually these data sets consist of a small number of experiments and are not amenable to other methods. Our approach is useful for small data sets for two reasons: First, we are using additional information in the form of a candidate set of motifs, so that even in a small expression data set where many genes have similar expression profiles, the motif information may provide enough evidence to resolve genes to meaningful clusters; second, the expression model exploits our prior belief that the "genetic trajectories" should be relatively smooth in time by using statistical splines [3], [14]. Therefore, all observations (time points) of a gene and all genes in a cluster contribute to estimates of the cluster-specific spline parameters. We note, however, that one could use a different gene expression probabilistic model to deal with other kinds of expression data (see, for example, Friedman [10]), and then apply the approach we outline here: Perform joint clustering on expression and motif data via EM to learn module assignments, and then use the JS entropy to characterize the most significant motifs associated to each module.

We present experimental results on the yeast cell cycle to validate our method. We find that our combined expression and motif clustering algorithm discovers coherent modules associated to known cell cycle transcription factors.

2 Joint Clustering Model for Motif and Expression Data

In order to learn "transcriptional modules"—that is, clusters of genes where similarity in expression is explained by a common regulatory mechanism at the transcriptional level—we perform a probabilistic assignment of genes to

modules based on two types of data for each gene i: The vector of expression values over the time course, $\mathbf{Y}_i$; and the sparse vector of motif counts corresponding to occurrences of regulatory elements in the promoter region of the gene, $\mathbf{R}_i$. The variable $\mathbf{Z}_i$ represents the "module" or cluster assignment of gene i. We assume a graphical model of the following form:

$$\mathbf{R} \rightarrow Z \rightarrow \mathbf{Y}.$$

Here, the gene expression (over a time series of microarray experiments) is conditioned on module assignment, and module assigment is conditioned on the presence of regulatory elements in the promoter sequence. The joint probability for this graphical model is $P(\mathbf{Y}, \mathbf{R}, Z) = P(\mathbf{Y}|Z)P(Z|\mathbf{R})P(\mathbf{R})$. Biologically, the model proposes that the presence of regulatory elements in the promoter sequence of a gene determines its module assigment and that, in a fixed time course, the module assignment explains the gene's expression profile.

However, by Bayes rule, we can rewrite the joint probability distribution to obtain the model

$$\mathbf{R} \leftarrow Z \rightarrow \mathbf{Y},$$

where $\mathbf{Y}_i$ and $\mathbf{R}_i$ are conditionally independent given module assignment Z_i:

$$P(\mathbf{Y}_i, \mathbf{R}_i, Z_i) = P(\mathbf{Y}_i|Z_i)P(\mathbf{R}_i|Z_i)P(Z_i).$$

We use this form of the probability distribution in our algorithm. The task is then to learn the parameters of the conditional probability models and the module assignments that maximize the likelihood for the input data, and our approach reduces to an expectation maximization-based joint clustering of the sequence and expression data.

For the sequence model, we start with a fixed set of known or putative motifs representing regulatory elements: $\mathbf{M} = \{m_1, m_2, \ldots, m_P\}$. Similar to the bag-of-words model used in text classification, we represent the promoter sequence for gene i as the sparse vector $\mathbf{R}_i$ of counts of motifs that it contains, where $\mathbf{R}_i$ is indexed by motifs in $\mathbf{M}$: $\mathbf{R}_i = (n_{i1}, n_{i2}, \ldots, n_{iP})$. We let $n_i = \sum_{p=1\ldots P} n_{ip}$ be the total count of motifs from $\mathbf{M}$ occurring in the promoter region for gene i. For each module j, we have a set a regulatory element frequencies $\Theta_j = (\theta_{j1}, \theta_{j2}, \ldots, \theta_{jP})$, where $\sum_{p=1\ldots P} \theta_{jp} = 1$. Our conditional probability model for the motif count data is a multinomial (or naive Bayes) model [18], where the events are seen as the counts n_{ip} and we sum over all sequences of motifs that can lead to the same set of counts:

$$P(\mathbf{R}_i|Z_i = j, \Theta_j) = \frac{n_i!}{n_{i1}! \ldots n_{iP}!} \Pi_{p=1\ldots P} \theta_{jp}^{n_{ip}}. \quad (1)$$

Here, the assumption is that motifs within the promoter sequence are generated independently of each other, according to the module-dependent motif emission frequencies Θ_j.

A few comments about the choice of naive Bayes model are in order. Ideally, the set of motifs provided to the algorithm—typically specified by position-specific scoring matrices or consensus patterns—would be filtered for redundancy, so that, in particular, one would not find occurrences of different motifs at the same or overlapping positions in a promoter sequence. In practice, it is difficult to eliminate redundancy entirely, and indeed subtly different motifs can encode different binding strengths or even correspond to different transcription factors. Moreover, binding sites for *different* transcription factors can in fact occur in overlapping positions in the promoter (for example in the case of competitive binding). It is therefore quite possible that the assumption of the naive Bayes model—that different "words" (motifs) are conditionally independent given the cluster assignment—could be violated for a particular set of motifs. However, it is also true that, even though the naive Bayes assumption is often violated in text data, the model is widely and successfully used for text processing tasks like document classification [18]. In the experiments described below, we therefore perform only conservative filtering on our initial set of motifs, and we rely on the clustering algorithm and the JS entropy-based motif scoring to determine which motifs are most relevant for each cluster.

To model time series expression data, we follow the approach of Bar-Joseph et al. [3] using statistical splines, motivated by the prior belief that genes have smooth expression trajectories and that genes whose trajectories have similar shapes may be under similar regulatory control. Each cluster is described by a Gaussian distribution over spline parameters that defines the general shape of the trajectories of its member genes. More precisely, we describe the true expression level for gene i, belonging to module j as a function of time as $(s_1(t) \ \cdots \ s_q(t))(\mu_j + \gamma_{ij})$, where $s_1(t), \ldots, s_q(t)$ are spline basis functions, μ_j is the mean vector of spline coefficients for module j, and γ_{ij} is the gene-specific variation vector of spline coefficients. We assume that each experimental observation is subject to Gaussian noise, with error $\epsilon \sim N(0, \sigma^2)$. Therefore, for a vector of observations for gene i at times $t_1, \ldots, t_m$, we have:

$$\mathbf{Y}_i = \begin{pmatrix} s_1(t_1) \cdots s_q(t_1) \\ \ddots \\ s_1(t_m) \cdots s_q(t_m) \end{pmatrix} \left[\begin{pmatrix} \mu_j^1 \\ \vdots \\ \mu_j^q \end{pmatrix} + \begin{pmatrix} \gamma_{ij}^1 \\ \vdots \\ \gamma_{ij}^q \end{pmatrix} \right] + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}.$$

The variable γ_{ij}, representing the variance from module expression mean μ_j, is most conveniently treated as a hidden variable. Here, γ_{ij} is normally distributed with mean $\mathbf{0}$ and covariance matrix Γ_j. The model for the conditional probability of gene expression, given module assignment, is then:

$$P(\mathbf{Y}_i, \gamma_{ij}|Z_i = j, \mu_j, \Gamma_j, \sigma^2) = (2\pi)^{-(m+q)/2} |\Gamma_j|^{-1/2} \sigma^{-m} \cdot e^{-1/2\sigma^2 (\mathbf{Y}_i - S(\mu_j + \gamma_{ij}))^t (\mathbf{Y}_i - S(\mu_j + \gamma_{ij}))} \quad (2)$$
$$e^{-\frac{1}{2}\gamma_{ij}^t \Gamma_j^{-1} \gamma_{ij}}.$$

We use expectation maximization (EM) to learn the module assigments. The EM updates for the spline expression models come from Bar-Joseph et al. [3]; we modify the algorithm by combining with sequence information using the model (1) above. Here, we use a pseudocount α for the multinomial model which, as we show below, will provide a way to control the relative importance of motif and expression data in the joint clustering algorithm.

Details of the EM updates and hyperparameter α are given in the Appendix.

While the EM algorithm converges to a local maximum of the complete log likelihood function, the likelihood surface has many poor local maxima and is sensitive to the initial values of the paramaters. We choose a starting point by performing k-means clustering on the expression data alone to set the initial cluster assignments $p(j|i)$ of gene i to cluster j. (See Section 4.1 for a discussion of model selection to determine an appropriate number of clusters.) A similar strategy of using a simple clustering algorithm to find a starting point for a more complex probabilistic clustering algorithm was used in Segal et al. [22] for learning transcriptional modules. Clearly, choosing a reasonable starting point still leads in general to convergence to a local rather than global maximum, but our statistical and biological validation (see Section 4.2), suggest we are reaching a consistent local maximum of high quality.

3 Entropy-Based Determination of Significant Motifs

Once we have performed EM-based clustering, we can use the motif parameters of the learned model to determine which motifs are markers for certain clusters and which motifs are coassigned to particular clusters. The distribution θ_{jp} suggests a natural quantitative measure of both, allowing discovery of the over and underrepresented motifs within a given cluster. A common reparameterization-invariant measure of the divergence in a set of distributions (θ_{jp}) produced by clustering, for which there is a prior probability p_j of being found in cluster j, is the Jensen-Shannon entropy [23], [15], $H_{JS}(\{\theta_{jp}\}) = H\left(\sum_j p_j \theta_{jp}\right) - \sum_j p_j H(\theta_j)$, where here $H(x) = -\sum_p x_p \log_2 x_p$. The expression may be rewritten

$$H_{JS} = \sum_j \sum_p p_j \theta_{jp} \ln_2(\theta_{jp}/\theta_p) \quad (3)$$

$$= \sum_j p_j D_{KL}[\theta_{jp}||\theta_p], \quad (4)$$

where $\theta_p = \sum_j p_j \theta_{jp}$ and D_{KL} is the Kullback-Leibler divergence (or *relative entropy*) [7]. The summand in (4) is an always-positive quantity which then can be used to quantify how exceptional a given cluster is. Moreover, the summand in (3), which is the contribution of an individual word to an individual cluster, gives a local (in p) measure of its importance to cluster assignment. In the experiments below, we use the rank ordering of JS entropy contributions $p_j \theta_{jp} \ln_2(\theta_{jp}/\theta_p)$ derived from the learned module parameters to produce a list of significant cluster-motif associations.

4 Experimental Results: Yeast Cell Cycle

We use time series expression data from Spellman et al. [9], who list a set of 799 genes of *Saccharomyces cerevisiae* found to be cell-cycle regulated. We eliminate genes that have more than two missing values, as well as those that have been eliminated from the Saccharomyces Genome Database (SGD), leaving 776 genes. We apply our method to the alpha-pheromone experiment, consisting of 18 time points sampled every 7 minutes and representing two full cell cycles. Note that this data set is *not* the cdc15 data set, whose data are a concatenation of even and odd time points.

We obtain the 500 bp 5′ promoter sequences of all *S. cerevisiae* genes from SGD. For each of these sequences, we search for transcription factor (TF) binding sites using the PATCH software licensed by TRANSFAC [26]. The PATCH tool uses a library of known and putative TF binding site motifs, some of which are represented by position specific scoring matrices and some by consensus patterns, from the TRANSFAC Professional database. Initially, all 532 binding site motif patterns given separate accession numbers by TRANSFAC are used as motif queries. Using the PATCH tool, we obtain counts of the number of copies of all query motifs in the promoter regions of the 776 select genes to create a motif matrix n_{ip}: The number of occurrences of motif p upstream of gene i. In order to reduce the influence of binding sites for generic transcription factors, which have very large counts, we remove all motifs p whose total number of occurrences is 0 or exceeds 1,000 in the set of 776 genes. These restrictions leave a smaller set of 306 motifs, which nonetheless suffer significant redundancy. For example, some consensus patterns given different accession numbers in TRANSFAC differ by only one character and lead to identical count statistics. Therefore, we group the original motifs into equivalence classes by asserting that two consensus pattern motifs are equivalent if they induce the same count statistics for the set of 776 promoter regions. In principle, motifs with identical count statistics need not be sequence similar, but it is essentially true for our data set. We obtain 247 equivalence classes of motifs, and we then represent each equivalence class by a single column in our motif matrix.

We implement our module learning algorithm in MATLAB. Our source code and supplementary data are publically available at http://www.cs.columbia.edu/compbio/module-clust.

4.1 Model Selection

Following Bar-Joseph et al. [3], we use natural cubic splines, where the size q of the spline basis is equal to the number of knots. We use evenly-spaced knots and choose $q = 9$ based on cross-validation for the expression-only version of the clustering algorithm (results not shown). Model selection for the number of clusters is performed using five-fold cross-validation by computing the log loss function of each trained model on each held out fold, which is given by the negative of the following likelihood function:

$$\text{likelihood} = \sum_{\text{held out genes } i} \log\left(\sum_{\text{clusters } j} p_j P(\mathbf{Y}_i|Z_i = j) P(\mathbf{R}_i|Z_i = j)\right).$$

Note that here,

$$P(\mathbf{Y}_i|Z_i = j) = P(\mathbf{Y}_i|Z_i = j, \mu_j, \Gamma_j, \sigma^2) \sim |\Sigma_j|^{-1/2} e^{-\frac{1}{2}(\mathbf{Y}_i - S\mu_j)\Sigma_j^{-1}(\mathbf{Y}_i - S\mu_j)^t},$$

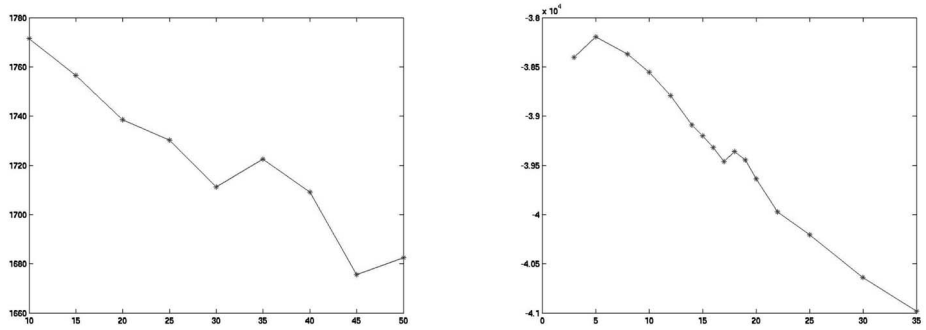


Fig. 1. Model selection by cross-validation. The figures plot the (a) cross-validation log likelihood for the model using expression data only with statistical splines and (b) for the combined expression and motif data model. A consistent (local) peak occurs at 18 clusters for the combined model.

where $\Sigma_j = \sigma^2 I + S\Gamma_j S^t$. Cross-validation results are shown in Fig. 1, using the expression data only with the statistical spline model (Fig. 1a) and using both expression data and promoter sequence data (Fig. 1b). While the log loss appears to be best for about 5 clusters in the combined model, we find a consistent local peak at 18 clusters; when we reran experiments with differently pruned versions of the motif matrix, we also consistently observe a peak at 18 or 19 clusters. This kind of phenomenon can occur when there are hierarchical clusters in the data, that is, a small number of large clusters that subdivide into smaller clusters. To investigate whether this is the case here, we considered all pairs of genes that belong to the same cluster in the 18-cluster model and found that 85 percent of these pairs also belonged to the same cluster in the 5-cluster model. Therefore, we see that the 18 smaller clusters are largely consistent with the 5 larger clusters; we choose the 18 smaller clusters for motif analysis because it is more biologically reasonable to have this number of modules with separate transcriptional control. We note that when using expression only or motif data only (not shown), there is no peak at 18 clusters, though the cross-validated log loss increases somewhat at 35 clusters in both cases. This suggests that by combining two sources of data, we may be discovering structure that we cannot learn on the basis of a single data type.

In combining two data types, we wish to understand the contribution of each data source to the overall probability model. A natural way to control the relative importance of motif and expression data is to scale the hyperparameter α for the multinomial model. When α is large—equivalent to a large pseudocount—the effect of the motif data is reduced, and cluster assignments become closer to those obtained using expression data alone; as α approaches 0, the motif data have a maximal influence on cluster assignments, but the expression data still contribute to the log likelihood function. Fig. 2 qualitatively illustrates some of the effects of changing the hyperparameter α. As α is decreased, increasing emphasis on motif data, the clustering algorithm is able to avoid learning a large cluster containing many of the genes with low expression levels; instead, by allowing the motif data to have more influence, many genes from the large expression cluster are reassigned to other clusters (Fig. 2a). Also, as α is decreased, the mean entropy of the multinomial distributions,

$$\frac{1}{J}\sum_{j=1}^{J} H(\Theta_j),$$

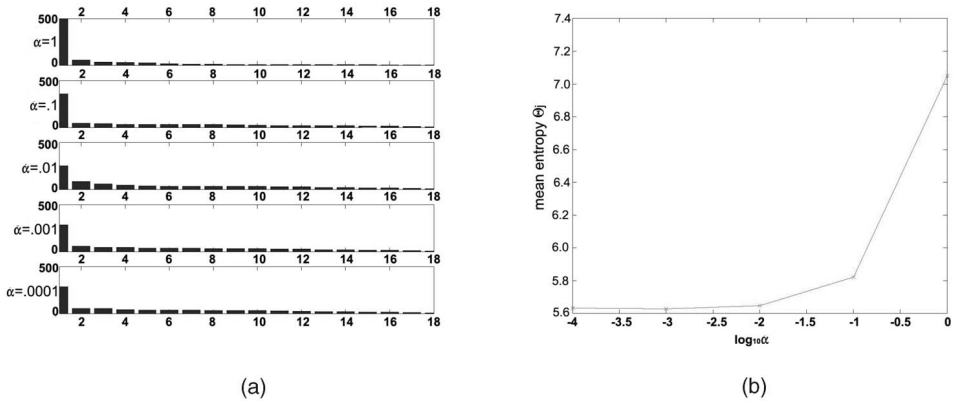


Fig. 2. Dependence on hyperparameter. (a) As hyperparameter α is decreased, increasing dependence on motif data, the model avoids learning a large cluster of genes with low expression levels and (b) the mean entropy of the multinomial motif models decreases. In both cases, the model is stable in the range $\alpha \in (0, 0.01]$.

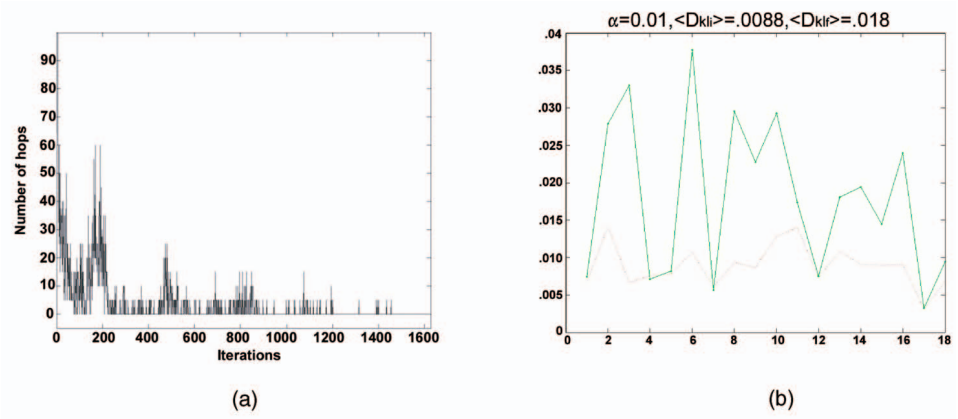


Fig. 3. Change of cluster assignments and Kullback-Leibler divergence for motif distributions. Plot (a) shows the number of cluster reassignments as a function of EM iteration. Plot (b) shows the Kullback-Leibler divergence of each cluster's motif distribution θ_{jp} relative to the average θ_p, at initial times (red) and final times (green).

where $H(\Theta_j) = -\sum_p \theta_{jp} \log_2(\theta_{jp})$ also decreases, indicating that "motif emission" random variables described by these distributions become less uncertain. We see that the model is stable when the hyperparameter is chosen in the range $(0, 0.01]$, that is, both the cluster size distributions and the mean entropy do not vary significantly in this range. Since we do not wish to deemphasize motif information, we choose α in the stable range ($\alpha = 0.01$) for a pseudocount. The use of hyperparameters for scaling the influence of different data sources has been proposed, for example, by Barnard et al. [5] in the case of integrating text and image data for learning from Web page data.

4.2 Model Validation and Interpretation

We first wish to establish that the EM algorithm converges to a solution significantly different from its starting point. We initialize the parameters for the EM algorithm by first performing k-means clustering on the expression data alone. However, many genes change from their initial cluster assignment during the course of the algorithm: We can quantify the rate of cluster reassignments by plotting the number of changes in the "hardened" $p(j|i)$ (resulting from assigning each gene with probability 1 to its cluster with maximum posterior $p(j|i)$) at each iteration, as shown in Fig. 3a).

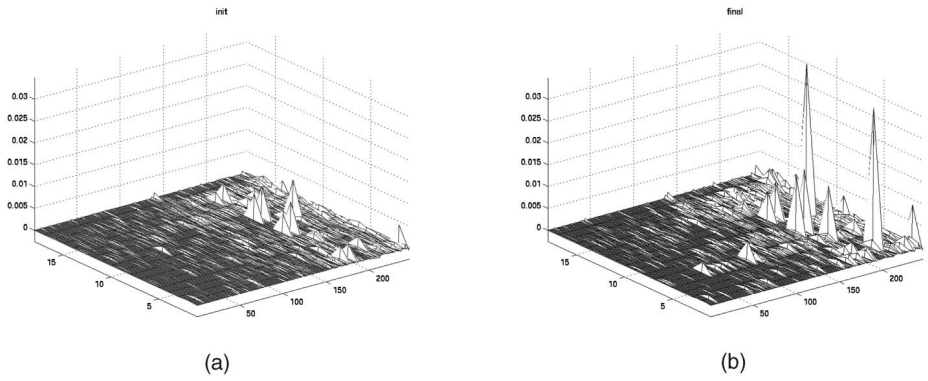
This function rapidly decays with less and less frequent rearrangements, and we may conclude with confidence that the EM procedure has converged after $\sim 1,500$ iterations (about 7 minutes of computation time on a 2.2 GHz Linux server).

It is also useful to quantify the success of the clustering in coassigning related motifs, and to discover which of the motifs are markers for certain cluster assignments. Recall that the summand in (4), $p_j D_{KL}[\theta_{jp}||\theta_p]$, is an always-positive quantity which can be used to quantify how exceptional a given cluster is. Reassuringly, we find that this quantity grows for all but three small clusters during the EM process (Fig. 3b), and in these three clusters the quantity stays the same or decreases negligibly. Since the initialization point for the EM algorithm is based on standard clustering of expression only, we can view this plot as a quantitative comparison our approach to the widely used method of clustering first and then looking for motifs: When we cluster using expression only, the distribution of motifs occurrences for different clusters less exceptional, that is, closer to the average distribution across all clusters.

Recall also that the summand in (3), $p_j \theta_{jp} \ln_2(\theta_{jp}/\theta_p)$, which is the contribution of an individual word to an individual cluster, gives a local (in p) measure of its importance to cluster assigment. Rank-ordering the weighted contributions of the Kullback-Leibler density to the JS entropy, $d_{jp} \equiv p_j \theta_{jp} \log_2(\theta_{jp}/\theta_p)$, gives a list of the individual motif-cluster combinations most important to the cluster assignments. The first four columns of Fig. 4 show the top 25 such combinations, including the TRANSFAC identifiers associated to each motif equivalence class p. TRANSFAC motifs that correspond to the same motif p are given the same numerical ranking in the first column of the table. Note that individual contributions to the JS entropy represent both over and underrepresented motifs (since d_{jp}, unlike $D_{KL}(\theta_{jp})$ or $H_{JS}(\{\theta_{jp}\})$, can be positive or negative), yet avoid the possibility of $\theta_{jp} \neq 0, \theta_p = 0$, regardless of the

rank	$p_j \theta_{jp} \ln(\theta_{jp}/\overline{\theta})$	cluster	motif ID	associated TF
1	0.0348155	10	Y$LEU2_03	TBP
2	0.0316795	3	Y$LEU2_03	TBP
3	0.0140682	8	Y$CDC9_02	DSC1
4	0.0128409	8	Y$POL1_02	MCBF
5	0.0120145	6	Y$CHA1_01	CHA4
6	0.00882799	2	Y$PHR1_01	RPH1 PRP
7	0.00794634	11	Y$CDC9_02	DSC1
8	0.00647105	11	Y$POL1_02	MCBF
9	0.00463384	6	Y$IPT1_01	PDR3 PDR1
10	0.00396947	9	Y$MAL61_04	MIG1
11	0.0038751	14	Y$MAL61_04	MIG1
12	0.0034978	16	Y$DAL3_01	DAL80
13	0.00333109	16	Y$GAL3_01	MIG1
14	0.0032351	16	Y$ARS1_05	ABF2
15	0.00283304	6	Y$CYC1_11	NF-YA HAP3
16	0.00272052	13	Y$MFAL1_03	MCM1 MATalpha1
17	0.00263832	14	Y$PHR1_01	RPH1 PRP
18	0.00247756	6	Y$SNQ2_02	PDR3
19	0.00241394	3	Y$CDC9_02	DSC1
20	0.00232386	2	Y$CHA1_01	CHA4
21	0.0022854	2	Y$TPI_01	GCR1
22	0.00225421	16	Y$CUP1_07	TBP
23	0.00218436	3	Y$POL1_02	MCBF
24	0.00203809	13	Y$MFAL1_04	MCM1
25	0.00202506	13	Y$TPI_01	GCR1

Fig. 4. Significant motifs for each cluster, ranked by weighted Kullback-Leibler density $d_{jp} \equiv p_j \theta_{jp} \log_2(\theta_{jp}/\overline{\theta})$, and the transcription factors that bind to them (as listed by TRANSFAC).



(a) (b)

Fig. 5. The weighted Kullback-Leibler density $d_{jp} \equiv p_j \theta_{jp} \log_2(\theta_{jp}/\overline{\theta})$, which can be positive (negative) for motifs over (under) represented in their cluster relative to other clusters. The positive values of the density are plotted here at (a) the initial time and (b) the final time. Note the extreme sharpening relative to the initialization before the EM procedure, showing that the learned modules display overrepresented motifs.

value of the pseudocount α, for which d_{jp} would be undefined. Note again that at the end of the EM algorithm, the density d_{jp} is extremely sharpened relative to the starting point, based on k-means clustering of expression data alone (Fig. 5).

In order to assist with biological validation, we also query the TRANSFAC motif records corresponding to the top scoring patterns in our list, noting which transcription factors (if any) are known to bind to these motifs. These results are listed in the last column of Fig. 4. Note that in the table, TRANSFAC motifs that we assign to the same equivalence class are indeed binding sites for the same transcription factor, which gives evidence that our grouping of motifs into equivalence classes was not overly aggressive. It is certainly true that not all motif redundancy was eliminated by our initial grouping into equivalence classes. Nonetheless, as we discuss in the following section, our method is able to identify meaningful associations of regulatory elements (and hence transcription factors, through the TRANSFAC annotations) to clusters.

4.3 Biological Validation

We evaluate our clusters for functionally coherent genes by looking for enrichment of Gene Ontology (GO) terms. For simplicity of analysis, we consider the hard cluster assignments obtained by assigning each gene to the cluster having maximum posterior probability. Using SGD's GO term finder, we find that 14 of our 18 clusters are enriched for annotations using a p-value threshold of 0.05. Cluster 1 consists of genes that peak in the M/G1 transition involved in the cytokinesis and cell proliferation. Cluster 2 is highly enriched for the subtelomerically encoded Y-helicases which peak in the G1 phase. The dominant binding sites are for CHA4, GLN3, DAL80, GAL4, SRF, MCM1, MATalpha2, ADR1, TBP, GCN4, and SKO1. Most of these genes have poorly understood functions. Cluster 3 is enriched for genes involved in energy reserve metabolism and several transcriptional repressors. It has dominant binding sites for the stress response regulators MSN2 and MSN4. Cluster 4 consists exclusively of histone genes that peak in the S phase of the cell cycle. These genes are involved in chromatin assembly and disassembly. The cluster was also enriched for binding sites of RPN4. Cluster 5 is enriched for binding sites of PHO4 which is required for the expression of the phosphate pathway. These genes are involved in phosphorus metabolism and their expression profiles peak at the G2/M transition. Genes in clusters 6 and 15 are involved in microtubule based processes. However, they have different expression and motif profiles. Cluster 6 peaks in the S and S/G2 phase, whereas cluster 15 peaks at the G1 phase. Cluster 15 is enriched for SUM1 binding sites which is involved in chromosomal segregation. Cluster 7 is made up of genes involved in mating, pheromone response and cell communication. Clusters 8 and 11 are made up of genes that peak in the G1 phase and are involved in the DNA replication process and initiation of the S-phase. These clusters are enriched for DSC1 (DNA synthesis control 1) binding sites which is a complex of CCBF/SBF(SWI4+SWI6) and CDC10 [16] and is mainly involved in regulation of DNA-synthesis genes [16], [17]. Cluster 10 was enriched for genes involved in methionine biosynthesis and sulphur metabolism. The dominant binding sites are those of MET32 and MET31. MET31 and MET32 are transcriptional regulators of sulfur amino acid metabolism [24]. Cluster 13 is made up of genes that peak in the G2/M transition and are mainly involved in steroid metabolism and ion transport. Cluster 14 has genes that peak at the G2/M transition and are involved in iron transport. The MAC1 binding site is significant for this cluster. It is involved in regulating copper and iron transport. Thus, we see a strong correlation between the functional enrichment of clusters using the GO and MIPS annotations and the dominant binding factors as assessed using the JS entropy. We also observe that the method is able to separate clusters with very similar expression profiles, but different motif profiles such as clusters 8 and 11. Cluster membership lists and transcription factors associated to the significant motifs for each cluster are given on the supplementary Web site (http://www.cs.columbia.edu/compbio/module-clust).

5 Conclusion

We have described a model for combining time series expression data and promoter sequence data for learning transcription modules in organisms—such as yeast—where there are already a large number of known motifs corresponding to regulatory elements. In experimental results on a yeast cell cycle data set, our learning algorithm finds a number of modules and associated binding site motifs for known cell cycle transcription factors with a functionally coherent and biologically plausible set of regulated genes.

We note that we cannot expect that combining data sources will always make clusters better defined by standard measures of coherence. For example, we should expect that expression coherence can in fact be lower when we incorporate motif data into the clustering, because genes with similar motifs could show different global expression profiles due to differences in combinatorial interactions of regulators at different stages of the cell cycle. Conversely, in some cases, genes having similar expression profiles need not have similar motif profiles, since different sets of transcription factors that are involved in regulating a particular biological processes can act in similar ways to produce similar expression patterns in the genes that they regulate.

Probabilistic generative models are a natural framework for integrating multiple data sources, once one has an appropriate representation of the individual data types and their probabilistic dependence on each other. Many extensions of our model are possible. For example, if one had prior biological knowledge about the importance or strength of particular motifs, one could encode this in the form of motif-specific parameter priors. One could also try to represent the binding activity of those known transcription factors whose binding motifs are also known, either as variables observed through gene expression profiles or as hidden variables.

The assumption of independence of motifs, used in the probabilistic model of promoter sequences, is clearly an oversimplification. In particular, we do not directly model cooperative or competing binding of transcription factors to spatially proximal site. However, in the original context of text applications, the assumption of word independence is similarly inaccurate, and yet the bag-of-words model works well and is widely adopted. Co-occurrence of motifs representing factors and cofactors can be investigated as a postprocessing step, by considering spatial proximity of the significant motifs for each cluster, or could be developed in the probabilistic model in future work. In particular, to scale up to more complex organisms, it would be appropriate to model the presence of cis regulatory modules—functional units consisting of spatially clustered groups of regulatory elements—rather than the presence of individual motifs. For example, to search for cis regulatory modules in *D. melanogaster* (fruit fly) associated with patterning in the early embryo, Rajewsky et al. [20] use a probabilistic segmentation of the regulatory sequence into binding sites and background using a fixed set of weight matrix motif models, where the probability of emitting each particular motif (or background sequence) is fit by expectation maximization. In a similar way, one could use such a probabilistic model as the sequence component of a joint clustering model with expression data, estimating cluster-specific motif emission probabilities using EM. Extending the approach to more complex organisms does, however, require some knowledge of true or putative binding site motifs, either experimentally verified or computationally determined. We believe that the relatively simple model presented here gives a promising starting point these extensions to richer modeling of combinatorial control and to regulation in more complex organisms.

APPENDIX

Here, we briefly outline details of the EM algorithm. For each gene i, both the module assignment Z_i and the gene-specific variation γ_{ij} from module expression mean μ_j are treated as hidden variables. We have the conditional probability model

$$P(\mathbf{R}_i, \mathbf{Y}_i, \gamma_{ij} | Z_i = j) = P(\mathbf{R}_i | Z_i = j, \Theta_j) P(\mathbf{Y}_i, \gamma_{ij} | Z_i = j, \mu_j, \Gamma_j, \sigma^2),$$

where $P(\mathbf{R}_i | Z_i = j)$ is given by (1) and $P(\mathbf{Y}_i, \gamma_{ij} | Z_i = j)$ is defined by (2).

In the E-step, we calculate

$$p(j|i) = \frac{p_j P(\mathbf{R}_i, \mathbf{Y}_i, \gamma_{ij} | Z_i = j, \Theta_j, \mu_j, \Gamma_j, \sigma^2)}{\sum_k p_k P(\mathbf{R}_i, \mathbf{Y}_i, \gamma_{ik} | Z_i = k, \mu_k, \Gamma_k, \sigma^2)},$$

where p_k are prior probabilities on the module assignments. Also in the E-step, we calculate expectations

$$\widehat{\gamma_{ij}} = (\sigma^2 \Gamma^{-1} + S^t S)^{-1} S^t (\mathbf{Y}_i - S\mu_j),$$
$$\widehat{\gamma_{ij}^t \gamma_{ij}} = \widehat{\gamma_{ij}}^t \widehat{\gamma_{ij}} + (\Gamma_j^{-1} + S^t S / \sigma^2)^{-1}.$$

In the M-step, we update parameters with

$$\sigma^2 = \frac{\sum_i \sum_j p(j|i) (\mathbf{Y}_i - S(\mu_j + \widehat{\gamma_{ij}}))^t (\mathbf{Y}_i - S(\mu_j + \widehat{\gamma_{ij}}))}{mN},$$

$$\Gamma_j = \frac{\sum_i p(j|i) \widehat{\gamma_{ij}^t \gamma_{ij}}}{\sum_i p(j|i)}, \qquad p_j = \frac{1}{N} \sum_i p(j|i)$$

$$\mu_j = \left(\sum_i p(j|i) S^t S \right)^{-1} \left(\sum_i p(j|i) S^t (\mathbf{Y}_i - S\widehat{\gamma_{ij}}) \right),$$

$$\theta_{jp} = \frac{\sum_i p(j|i) n_{ip} + \alpha}{\sum_p \sum_i p(j|i) n_{ip} + \alpha P}.$$

Here, the α functions as a pseudocount or, more generally, as a hyperparameter for the Dirichlet prior for the multinomial parameters θ_{jp}. (We use the same pseudocount for all motif frequency parameters in all modules.)

ACKNOWLEDGMENTS

The authors thank Mark Johnson, Ilya Nemenman, Tony Jebara, and William Stafford Noble for helpful technical discussions and suggestions. Anshul Kundaje is supported by US National Science Foundation grant EEC-00-88001. Chris Wiggins and Manuel Middendorf are partially supported by US National Science Foundation grants ECS-0332479 and NIH GM36277. Christina Leslie and Chris Wiggins are supported by NIH grant LM07276-02, and Christina Leslie is supported by an Award in Informatics from the PhRMA Foundation. Complete experimental results and MATLAB implementation are available at http://www.cs.columbia.edu/compbio/module-clust.

REFERENCES

[1] T.L. Bailey and C. Elkan, "Unsupervised Learning of Multiple Motifs in Biopolymers Using EM," *Machine Learning*, vol. 21, nos. 1-2, pp. 51-80, 1995.

[2] L.D. Baker and A.K. McCallum, "Distributional Clustering of Words for Text Classification," *Proc. SIGIR-98, 21st ACM Int'l Conf. Research and Development in Information Retrieval*, pp. 96-103, 1998.

[3] Z. Bar-Joseph, G. Gerber, D.K. Gifford, T.S. Jaakkola, and I. Simon, "A New Approach to Analyzing Gene Expression Time Series Data," *Proc. RECOMB Conf.*, 2002.

[4] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan, "Matching Words and Pictures," *J. Machine Learning Research,* vol. 3, pp. 1107-1135, 2003.
[5] K. Barnard and D. Forsyth, "Learning the Semantics of Words and Pictures," *Proc. Int'l Conf. Computer Vision,* vol. 2, pp. 408-415, 2001.
[6] M.A. Beer and S. Tavazoie, "Predicting Gene Expression from Sequence," *Cell,* vol. 117, no. 2, pp. 185-98, Apr. 2004.
[7] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* New York: John Wiley, 1990.
[8] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Nat'l Academy of Sciences,* vol. 95, pp. 14863-14868, 1998.
[9] T.S. Spellman et al., "Comprehensive Identification of Cell Cycle-Related Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization," *Molecular Biology of the Cell,* vol. 9, pp. 3273-3297, 1998.
[10] N. Friedman, "PCluster: Probabilistic Agglomerative Clustering of Gene Expression Profiles," technical report, Stanford Univ., 2003.
[11] I. Holmes and W.J. Bruno, "Finding Regulatory Elements Using Joint Likelihoods for Sequence and Expression Profile Data," *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology,* pp. 202-210, 2000.
[12] J.D. Hughes, P.W. Estep, S. Tavazoie, and G.M. Church, "Computational Identification of Cis-Regulatory Elements Associated with Groups of Functionally Related Genes in Saccharomyces Cerevisiae," *J. Molecular Biology,* vol. 296, no. 5, pp. 1205-1214, 2000.
[13] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing Modular Organization in the Yeast Transcriptional Network," *Nature Genetics,* vol. 31, pp. 370-377, 2002.
[14] G. James and T. Hastie, "Functional Linear Discriminant Analysis for Irregularly Sampled Curves," *J. Royal Statistical Soc.,* 2001.
[15] J. Lin, "Divergence Measures Based on the Shannon Entropy," *IEEE Trans. Information Theory,* vol. 37, pp. 145-151, 1991.
[16] N.F. Lowndes, A.L. Johnson, L. Breeden, and L.H. Johnston, "Swi6 Protein is Required for Transcription of the Periodically Expressed DNA Synthesis Genes in Budding Yeast," *Nature,* vol. 357, pp. 505-508, 1992.
[17] N.F. Lowndes, A.L. Johnson, and L.H. Johnston, "Coordination of Expression of DNA Synthesis Genes in Budding Yeast by Cell-Cycle Regulated Trans Factor," *Nature,* vol. 350, pp. 247-250, 1991.
[18] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *Proc. AAAI-98 Workshop Learning for Text Categorization,* 1998.
[19] Y. Pilpel, P. Sudarsanam, and G.M. Church, "Identifying Regulatory Networks by Combinatorial Analysis of Promoter Elements," *Nature Genetics,* vol. 2, pp. 153-159, 2001.
[20] N. Rajewsky, M. Vergassola, U. Gaul, and E.D. Siggia, "Computational Detection of Genomic CIS Regulatory Modules, Applied to Body Patterning in the Early Drosophila Embryo," *BMC Bioinformatics,* vol. 3, no. 30, 2002.
[21] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module Networks: Discovering Regulatory Modules and Their Condition Specific Regulators from Gene Expression Data," *Nature Genetics,* vol. 34, no. 2, pp. 166-176, 2003.
[22] E. Segal, R. Yelensky, and D. Koller, "Genome-Wide Discovery of Transcriptional Modules from DNA Sequence and Gene Expression," *Bioinformatics,* vol. 19, 2003.
[23] N. Slonim, N. Friedman, and N. Tishby, "Agglomerative Multivariate Information Bottleneck," *Proc. Neural Information Processing Systems Conf. (NIPS-12),* pp. 617-623, 2000.
[24] D. Thomas and Y. Surdin-Kerjan, "Metabolism of Sulfur Amino Acids in Saccharomyces Cerevisiae," *Microbiology and Molecular Biology Rev.,* vol. 61, pp. 503-532, 1997.
[25] G.C. Tseng, M.-K. Oh, L. Rohlin, J.C. Liao, and W. Wong, "Issues in cDNA Microarray Analysis: Quality Filtering, Channel Normalization, Models of Variations and Assessment of Gene Effects," *Nucleic Acids Research,* vol. 29, no. 12, pp. 2549-2557, 2001.
[26] E. Wingender, X. Chen, R. Hehl, H. Karas, I. Liebich, V. Matys, T. Meinhardt, M. Prüss, I. Reuter, and F. Schacherer, "TRANSFAC: An Integrated System for Gene Expression Regulation," *Nucleic Acids Research,* vol. 28, pp. 316-319, 2000.

Anshul Kundaje received the BS degree from Veermata Jijabai Technological Institute (VJTI) at the University of Mumbai in 2001 and the MS degree from Columbia University in 2002, both in electrical engineering. He is pursuing the PhD degree in computer science at Columbia University. His research focus is computational biology, specifically the application of machine learning techniques and statistical methods to analyze hard biological problems. He is primarily interested in microarray noise analysis, regulatory motif discovery and learning regulation networks using multiple sources of high-throughput biological data.

Manuel Middendorf received the undergraduate degree in physics from the Technical University in Berlin, Germany. He is currently pursuing the PhD degree in physics from Columbia University. His research focus is the use of machine learning and information theory for analyzing and predicting properties of biological networks.

Feng Gao received the medical degree from Beijing University in 1992 and the PhD degree in molecular cell biology from the Medical College of Ohio in 2000. He has held postdoctoral positions in computational biology at the University of Michigan, Ann Arbor, and at Columbia University.

Chris Wiggins received the PhD degree in theoretical physics from Princeton University in 1998. He is now an assistant professor of applied mathematics at Columbia University.

Christina Leslie received the PhD degree in mathematics from the University of California at Berkeley in 1998. She held appointments as a postdoctoral scientist in mathematics and then as an assistant professor of computer science at Columbia University before joining the Center for Computational Learning Systems, also at Columbia, where she is currently a research scientist. Her research focuses on applying machine learning methods to problems in computational biology, including protein sequence analysis, modeling gene regulatory networks, and computational analysis of pre-mRNA splicing. She leads the Computational Biology Group at Columbia University and is a member of Columbia's Center for Computational Biology and Bioinformatics (C2B2). She is a member of the IEEE Computer Society.