Machine learning methods for transcription data integration

D. T. Holloway
M. A. Kon
C. DeLisi

Gene expression is modulated by transcription factors (TFs), which are proteins that generally bind to DNA adjacent to coding regions and initiate transcription. Each target gene can be regulated by more than one TF, and each TF can regulate many targets. For a complete molecular understanding of transcriptional regulation, researchers must first associate each TF with the set of genes that it regulates. Here we present a summary of completed work on the ability to associate 104 TFs with their binding sites using support vector machines (SVMs), which are classification algorithms based in statistical learning theory. We use several types of genomic datasets to train classifiers in order to predict TF binding in the yeast genome. We consider motif matches, subsequence counts, motif conservation, functional annotation, and expression profiles. A simple weighting scheme varies the contribution of each type of genomic data when building a final SVM classifier, which we evaluate using known binding sites published in the literature and in online databases. The SVM algorithm works best when all datasets are combined, producing 73% coverage of known interactions, with a prediction accuracy of almost 0.9. We discuss new ideas and preliminary work for improving SVM classification of biological data.

Introduction

A first step in understanding transcriptional regulation requires the mapping of proteins called transcription factors (TFs) to the genes they regulate and to the particular nucleotide sequences to which they bind. Typically, TFs bind to sites that are 10–15 nucleotides (nt) in length. Even a cursory examination of the DNA sequences that bind a particular TF indicates that the sequences are not identical, but instead define a motif, or similar pattern of nucleotide bases. The set of sites to which a particular TF binds will provide the basic input for computational methods that can be used to find additional sites.

These computational methods fall into two broad categories: supervised and unsupervised. The former starts with two example sets of potential TF-target sequences, each of which often consists of several hundred bases that are upstream from the potential target genes. The sequences are those known to bind a particular TF (called "positives") and those known not to bind ("negatives"), both of which are used to derive a classification rule using an SVM or other learning algorithm. Unsupervised methods begin with sets that are believed, on the basis of independent evidence, to contain a characteristic but unknown nucleotide pattern, which may represent a binding site. A search algorithm such as Gibbs sampling can be used to identify such a pattern within the promoter region of the gene (the part of the gene that is upstream from the exon or coding region of DNA). The promoter region binds RNA polymerase and transcription factors in order to begin transcription. Many unsupervised techniques for predicting binding sites have been explored in the literature [1–8], and an excellent review of current motif discovery and pattern analysis methods is available [9].

Our approach is meant to easily combine a large number of data types in a supervised learning scheme to more accurately predict the association of a transcription

0018-8646/06/$5.00 © 2006 IBM

factor and its targets. There are a number of ways to proceed, including methods that involve support vector machines (SVMs) and Bayesian variants, and approaches that use weighted combinations of both of these methods. In order to assign weights, we must know how well each method performs. Here we study the use of support vector machines, which we show can easily accommodate high-dimensional genomic datasets containing many hundreds or thousands of features. An example of such a dataset is a gene expression profile, with each gene described by hundreds of expression measurements taken under different conditions. These measurements are the dimensions (features) used to derive a classification rule. We also study a simple SVM framework for combining heterogeneous and diverse data.

The performance of supervised approaches is often reported in terms of a few basic statistics. Known positives and negatives are first divided into training and test sets. This can be done once, as in a "hold-out" method, or the division can be performed randomly many times, as in a "cross-validation" procedure. In either case, the algorithm learns, using the training set, and makes predictions on the test set. Correct positive predictions (TP, "true positives"), correct negative predictions (TN, "true negatives"), incorrect positive predictions (FP, "false positives"), and incorrect negative predictions (FN, "false negatives") are counted for each test set and used to calculate more informative measures of performance. Two simple measures are sensitivity (S), which is the percentage of known targets correctly predicted to be true, and positive predictive value [$PPV = TP/(TP + FP)$], the percentage of positive predictions that are correct. Other measures are also possible, as we discuss below.

A number of supervised approaches have been used to associate transcription factors with their targets. Original work in transcription-factor binding-site discovery involved the use of position-specific scoring matrices (PSSMs) [10–13], which record the frequency of nucleotide bases at each position in a binding-site representation, or motif. A new prediction is then a site that matches the PSSM on the basis of a score threshold [10]. Researchers subsequently discovered that clusters of predicted binding sites can indicate whether a candidate gene is a target of a regulator [14–17]. Another supervised method, developed by the team of N. Simonis at the Centre de Biologie Structurale et Bioinformatique in Belgium, makes use of linear discriminant analysis (LDA) to select from a set of potentially co-regulated genes that are likely to share transcription factors. Using a set of 1,012 regulatory interactions involving 66 TFs [data obtained from the Transcription Factor Database (TRANSFAC**) [18], the aMAZE database [19], and a list compiled by Young et al. [20] from the Yeast Proteome Database], the researchers report an average positive predictive value of 0.91 and a sensitivity of 73%. Their classification performance based on ChIP-chip (Chromatin Immuno-Precipitation microarray) data is worse, with only 52% of genes identified by ChIP-chip being discovered. ChIP-chip is a large-scale procedure designed to experimentally identify transcription-factor targets genome-wide [21]. A microarray error model determines the significance of the identified targets. The Simonis team has argued in the past that ChIP-chip results likely contain many false positives; however, their results also show that target groups identified by ChIP experiments contain large numbers of motifs that are significantly overrepresented in comparison to random gene sets. This suggests that many of the targets generated by high-throughput experiments, such as chromatin immunoprecipitation, contain real binding-site signals.

In an approach more closely related to ours, Qian et al. apply support vector machines to gene expression profiles in order to predict TF–target relations [22]. Gene expression profiles are simply vectors, one for each gene, whose components are measurements of the expression level of the gene under different conditions. Positive examples for the classifier are known TF–target pairs; negatives are randomly chosen relations. In a method that differs from ours, Qian et al. create one classification rule covering all TFs and targets, while in our method a classifier is constructed for each TF individually. In their formulation, the data for each known TF–target association is given as a concatenation of the TFs and the target's expression vectors over 79 experimental conditions (giving a 158-element vector to describe a positive example of regulation). Negatives are constructed similarly for genes chosen randomly and for those found to lack a TF binding site. Their best reported accuracy is 0.93; however, this result is somewhat misleading because their analysis contains a total of only 175 positives. Their classification of a large negative set (1,750 negatives) can result in high accuracy because large numbers of negatives are classified correctly. To put their result in perspective, their sensitivity is 55% and their positive predictive value is 63%. While their method shows promise, it still relies only on the correlation of the expression of the transcription factor to its target. Thus, they are likely to miss interactions depending on cooperating TFs, or factors whose activation is dependent on post-translation modification or nuclear exclusion.

The approach by Beer and Tavazoie uses Bayesian networks to learn the combinatorial relationships of TFs and targets that underlie gene expression data [23]. Their method begins by clustering gene expression data by similarity of expression and then using hierarchical Bayesian networks to predict the cluster assignment of a test gene on the basis of the sequences in its promoter.

They impose constraints, which can be learned by the algorithm, allowing them to derive complex logical relationships from the data (e.g., motif A and motif B must both be present and within 20 base-pairs). Although this approach is innovative and can accurately describe the sequence/expression relationships of many genes, it may not be appropriate for our goals because it can depend on the clustering of the expression data and the method by which motif discovery is performed on the genes being tested.

Our approach uses SVMs to associate TFs with targets by combining high-dimensional heterogeneous datasets, building on our previous work, which used fewer data sources [24]. SVMs have been applied successfully to many problems in computational biology. They have been used for the prediction of protein remote homology [25], secondary structure [26], protein subcellular localization [27], signal peptide cleavage sites [28], normal or cancerous tissue types [29], gene function [30], mRNA splice sites, and translation start sites [31]. One notable attempt combines information on protein sequence similarity, protein–protein interactions, protein hydrophobicity, and gene expression to predict the function of a set of proteins [32].

Background and brief review

We now introduce our methodology for the non-specialist and briefly review some basic elements of SVM algorithms. We have trained an SVM on each of 104 transcription factors (i.e., "regulators") independently, using positive and negative training sets as explained in the following paragraphs. Each gene in the positive set shares certain attributes, or features, that other genes do not share, and it is on the basis of these that a classifier for a particular TF is obtained. We use 18 different genomic datasets to generate attributes, as indicated in the following—for example, the number of occurrences of a particular nucleotide sequence of length k. For such a dataset, the number of occurrences of each of the 256 possible nucleotide sequences of length 4 ("4-mers") might be represented by a 256-component vector (a "feature vector"), each component of which is the number of times the corresponding 4-mer occurs upstream from one of the genes in the set. (In molecular biology, the term "upstream" refers to a relative position along the DNA or RNA sequence and denotes a region toward the 5′ end of the sequence.) To construct a classifier, positive examples (feature vectors of promoters known to be bound by a TF) and negative examples (those of promoters known not to be bound) must be identified. Given this set of data, each example is represented by a feature vector of attributes. In the case of k-mer counts, the components of the feature vector are counts of different k-mers that appear in the promoter region. Other datasets for the same TF will have the same example target genes, represented by different feature vectors. For example, a phylogenetic profile vector, which shows the occurrence of an ortholog, or ancestry-related sequence, in a set of 65 genomes, would be a vector of length 65 consisting of binary numbers, with 1 indicating the presence of an ortholog and 0 indicating its absence. Thus, the data for any particular TF consists of a number of different feature vectors in spaces with possibly thousands of dimensions (attributes), each such vector representing a gene in the training set.

The SVM algorithm separates the positive and negative sets in the feature space by finding a hyperplane whose distance from the closest data points of each class is maximal. Two parallel hyperplanes that pass through these closest data points are found, and a separator bisects the distance between them. Better generalization (i.e., performance in prediction) can be obtained by forgoing perfect separation of training data and allowing some misclassification. This *soft margin* SVM finds the hyperplanes under the constraint that the distance to the closest cleanly separated data be maximal, with some penalty for misclassifications, as explained below.

We denote the feature vector–output pair for the ith gene in the training set by $(\mathbf{x}_i, y_i)$, with y_i equal to $+1$ when $\mathbf{x}_i$ is a feature vector from the positive set, and -1 otherwise. The vector $\mathbf{x}_i$ has the form $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \cdots, x_{id})$, where d is the number of features, i.e., the dimensionality of the feature space.

The separating hyperplane H has the form

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{1}$$

(see **Figure 1**), in which the components of $\mathbf{w} \equiv (w_1, w_2, \cdots, w_d)$ are the weights of the corresponding features, with $b/|\mathbf{w}|$ representing the distance from the origin to the closest point on H.

For clarity, we first describe the case in which the positive and negative examples in feature space are completely separable by a hyperplane; we then discuss the nonseparable case that allows for misclassification. The challenge in the separable case is to find the values of $\mathbf{w}$ and b that give the maximum margin separating hyperplane (the one that separates the two classes most widely). This requires the use of only the closest correctly separated feature vectors, each representing the attributes of a gene. In the simple two-dimensional example in Figure 1, the feature vectors are $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$; note that the separator bisects the distance between parallel planes through those points. This (separable) situation is illustrated in Figure 1, but without the (misclassified) vector $\mathbf{x}_4$.

The margin is the distance between two planes parallel to the separator, one passing through the closest correctly classified positive data point, and the other passing

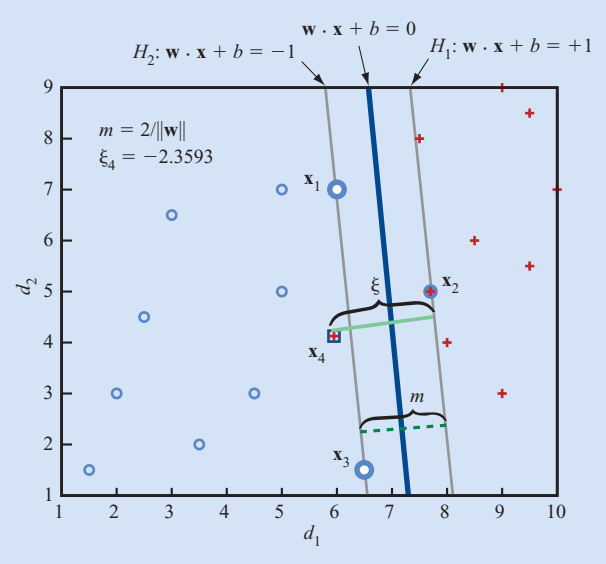

Figure 1

Anatomy of an SVM in two dimensions; this is the classification plot for the data given in Table 1. Red crosses (+) indicate positive examples and blue circles (o) are negatives. Coordinates d_1 and d_2 are the components of $\mathbf{x}$. The labeled points $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$ are the support vectors. The classifier is labeled as $\mathbf{w} \cdot \mathbf{x} + b = 0$, and the margin is labeled m. One point, $\mathbf{x}_4$, is misclassified. Because $\mathbf{x}_4$ is in the positive set, its slack variable ξ_4 is the distance from the $+1$ margin line.

through the closest correctly classified negative data point. The vector $\mathbf{w}$ is scaled so that the hyperplanes through the closest data (the *support vectors*) are given [33, 34] by $\mathbf{w} \cdot \mathbf{x} + b = +1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$.

Equivalently, the data satisfy the single constraint

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{2}$$

because $y_i(\mathbf{w} \cdot \mathbf{x}_i + b)$ is the distance from the separator to the ith data point. The margin (the perpendicular distance between the hyperplanes H_1 and H_2 parallel to H) is

$$m = \frac{2}{\|\mathbf{w}\|}, \tag{3}$$

where $\|\mathbf{w}\| = \sum_i w_i^2$ is the magnitude of the weight vector. Equation (3) is readily obtained by noting that if $\mathbf{x}^+$ and $\mathbf{x}^-$ denote the position vectors of two points at the intersection of an orthogonal to the separator with the margin hyperplanes ($\mathbf{x}^+$ and $\mathbf{x}^-$ can both be chosen parallel to $\mathbf{w}$; see Figure 1) and taking without loss $\|\mathbf{x}^+\| > \|\mathbf{x}^-\|$, then

$$m = \|\mathbf{x}^+ - \mathbf{x}^-\| = \|\mathbf{x}^+\| - \|\mathbf{x}^-\|.$$

On the other hand, from Equation (2), because $\mathbf{w}$ and $\mathbf{x}^\pm$ have been chosen to be parallel, we have $y_i(\|\mathbf{w}\| \cdot \|\mathbf{x}^\pm\| + b) = 1$ (using $\mathbf{x}^\pm$ with $y_i = \pm 1$, respectively), from which Equation (3) follows from the above, using

$$\|\mathbf{x}^+\| = \frac{-b+1}{\|\mathbf{w}\|}$$

and

$$\|\mathbf{x}^-\| = \frac{-b-1}{\|\mathbf{w}\|}.$$

Thus the problem is to maximize m given by Equation (3), subject to the constraints given by Equation (2). Note that in Equation (2), equality holds exactly for the support vectors, a subcollection that we label $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_s$, so that we can reduce the problem so that it involves this set of vectors only. The constrained maximization can be solved using Lagrange multipliers [33, 34]. In particular, the challenge is to minimize

$$L = \frac{\|\mathbf{w}\|^2}{2} - \sum_i \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]. \tag{4}$$

The weight vector is obtained by setting $(\partial L/\partial \mathbf{w}) = 0$ [i.e., the system of equations $(\partial L/\partial w_i) = 0$] for the extremal $\mathbf{x}_i$, i.e., those for which equality holds in Equation (2). These are exactly the support vectors, giving

$$\mathbf{w} = \sum_{i=1}^{s} \alpha_i y_i \mathbf{x}_i, \tag{5}$$

where, in this example, the weight vector is $\mathbf{w} \equiv (w_1, w_2)$, the ith attribute vector is $\mathbf{x}_i \equiv (x_{i1}, x_{i2})$, and the number of support vectors, s, is three (those lying on the margin planes). The one misclassified point $\mathbf{x}_4$ is currently ignored, but also becomes a support vector when included in the data, as described shortly.

The parameter b is determined as a weighted average of the distances to the two hyperplanes containing the support vectors,

$$b = \frac{1}{s} \sum_{i=1}^{s} y_i - \mathbf{w} \cdot \mathbf{x}_i. \tag{6}$$

In fact, in this fully separated case, for each support vector $\mathbf{x}_i$, $i = 1, \cdots, s$, we have $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$, by definition. The procedure for finding the multipliers α_i is somewhat simplified by forming and then maximizing the so-called dual Lagrangian [33, 34]

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{7}$$

This is obtained by substituting Equation (5) into Equation (4) and noting that $(\partial L/\partial b) = 0$ implies $\sum_i \alpha_i y_i = 0$.

examples	features		labels	Lagrange		calculate w vector			calc b		
	d 1	d 2	yi	alpha		$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$			$b^{(r)} = y_r - \mathbf{w} \cdot \mathbf{x}_r$	Distance to hyperplane	Slack variables ξ
1	6	7	-1	3.3793	Support vector	-20.2757	-23.655		-9.7425	-1	
2	7.7	5	1	1.7731	Support vector	13.6529	8.8655		-9.7425	1	
3	6.5	1.5	-1	3.3938	Support vector	-22.0598	-5.0907		-9.7425	-1	
4	6	4	1	5	misclassified	30	20			-1.3593	-2.3593
5	9	3	1	0		0	0			2.4731	
6	9.5	8.5	1	0		0	0			3.7904	
7	10	7	1	0		0	0			4.2695	
8	9	9	1	0		0	0			3.1916	
9	8.5	6	1	0		0	0			2.1737	
10	7.5	8	1	0		0	0			1.0958	
11	9.5	5.5	1	0		0	0			3.4311	
12	4.5	3	-1	0		0	0			-3.4551	
13	2.5	4.5	-1	0		0	0			-5.9102	
14	5	7	-1	0		0	0			-2.3174	
15	1.5	1.5	-1	0		0	0			-7.5868	
16	2	3	-1	0		0	0			-6.7485	
17	3.5	2	-1	0		0	0			-4.8922	
18	3	6.5	-1	0		0	0			-5.012	
19	5	5	-1	0		0	0			-2.5569	
20	8	4	1	0		0	0			1.2755	
					$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$	1.3174	0.1198	*mean(b)=*	-9.7425		

Figure 2

Excel** spreadsheet showing data and parameters of the classifier.

We now describe the case of soft margins, which is the formulation we use in practice where perfect separation is not possible (consider Figure 1 with the misclassified $\mathbf{x}_4$ now included). In this case, a penalty ξ_i is paid in the Lagrangian for each misclassification of size ξ_i (the distance of the misclassified $\mathbf{x}_i$ from its margin; see Figure 1). The target function and constraints are now modified so that the problem is to find

$$\min_{\mathbf{w},\xi} ||\mathbf{w}||^2 + C\sum_{i=1}^{r} \xi_i ,$$

subject to $\xi_i \geq 0$ and $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \cdots, r$.

As mentioned, ξ_i is the distance of the ith misclassified point $\mathbf{x}_i$ from its margin, which is defined as before to be the hyperplane (H_1 or H_2) at a distance $(1/||\mathbf{w}||)$ in the direction of the correct classification of $\mathbf{x}_i$ from the separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$. Parameter C mediates the tradeoff between maximal margin and misclassification, and r is the number of misclassified points allowed. Essentially, the algorithm proceeds to find the maximum margin by minimizing $||\mathbf{w}||$ while balancing this against the amount $\sum_{i=1}^{r} \xi_i$ of misclassification with this choice of margin. We again use Lagrange multipliers α_i as in the previous case to form a full Lagrangian, and then minimize it.

To illustrate in our two-dimensional example, we make use of the data presented in **Figure 2**. In this case, we then have, after minimizing the Lagrangian,

$$w_1 = \sum_i \alpha_i y_i x_{i1} = (1.7731)(1)(7.7) + (3.3938)(-1)(6.5) + (3.3793)(-1)(6) = 1.3174;$$

$$w_2 = \sum_i \alpha_i y_i x_{i2} = (1.7731)(1)(5) + (3.3938)(-1)(1.5) + (3.3793)(-1)(7) = 0.1198.$$

Thus $\mathbf{w} = [1.3174, 0.1198]$, and $b = [1 - (1.3174)(6) - (0.1198)(7) + \cdots]/3 = -9.7425$.

For linearly separable data (no misclassifications), we have $\xi_i = 0$, and we are in the first case, in which the values of $\mathbf{w}$ and b ensure that Equation (3) is minimized subject to the constraint of Equation (2). However, for data that is not linearly separable (e.g., including $\mathbf{x}_4$), α_i can become extremely large. In the Lagrangian formalization, a constant C becomes an upper bound for α_i (i.e., the constraint $0 \leq \alpha_i \leq C$ is used). By using C as a bound for α_i, we can limit the influence of single data points that cannot be classified correctly. Thus, in our example with $C = 5$, the multiplier for $\mathbf{x}_4$ has a value of 5 (Figure 2).

It is evident from Equation (7) that the first step in finding this maximal margin separator requires the calculation of all pairwise correlations between example vectors in the form of their inner (dot) products (also called the linear kernel function). Thus, given two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ yields a complete kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ involving

Table 1 Common kernel functions.

Kernel	*Parameters*	*Description*
Linear	None	$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$
Polynomial	Poly degree d	$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d$
Gaussian radial basis function (RBF)	σ	$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\lvert\mathbf{x} - \mathbf{y}\rvert^2}{2\sigma^2}\right)$
Gaussian	σ	$K(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

Table 2 Dataset abbreviations and description.

	Abbreviation	*Description*
1	MOT	Motif hits in S. *cerevisiae*
2	CONS	Motif hits conservation 18 organisms
3	PHY	Phylogenetic profile
4	EXP	Expression correlation
5	GO	GO term profile
6	KMER	*k*-mers – 4, 5, 6-mers
7	S1	Split 6-mer 1 gap kkk_kkk
8	S2	Split 6-mer 2 gaps kkk__kkk
9	S3	Split 6-mer 3 gaps kkk___kkk
10	S4	Split 6-mer 4 gaps kkk____kkk
11	S5	Split 6-mer 5 gaps kkk_____kkk
12	S6	Split 6-mer 6 gaps kkk______kkk
13	S7	Split 6-mer 7 gaps kkk_______kkk
14	S8	Split 6-mer 8 gaps kkk________kkk
15	MM01	6-mer with one mismatch (count 0.1)
16	MM05	6-mer with one mismatch (count 0.5)
17	ENT	Condition-specific TF–target correlation
18	SPAR	Nucleotide sparse binary encoding

every pair of data points. Because the data are represented internally only as such inner products rather than as explicit feature vectors, it becomes possible and useful to substitute different definitions of the inner product for the above linear dot product. Several alternatives are given in **Table 1**. These functions are inner products defined on feature spaces of different dimensionalities. Defining such a new inner product implicitly maps the data into a new feature space. This swapping of kernel functions in order to map data into different spaces is commonly referred to as the "kernel trick." Biological features such as conservation or gene expression values can be correlated and may have complex, nonlinear relationships, highlighting the need for classification schemes that can accurately classify data that are not linearly separable.

Datasets

We have tested a variety of sequence- and non-sequence-based classifiers for predicting the association of TFs and genes. All together, 18 separate data sources (each yielding a feature map and kernel) are combined to build classifiers for each transcription factor. The 18 data sources comprise a family of sequence-based methods (e.g., *k*-mer counts and TF motif conservation in multiple species), expression datasets, phylogenetic profiles, and gene ontology (GO) functional profiles (see **Table 2**). For a more detailed description of datasets, see [35]. In almost all cases, our datasets have complete information, primarily because datasets such as *k*-mer counts or motif counts are derived from DNA sequences alone. Microarray expression data is also available for every gene in our analysis. In the cases in which expression values are missing for a few conditions, zeros are substituted, as is often done in computational analyses. For the GO functional profiles and the phylogenetic profiles based on the Cluster of Orthologous Groups (COG) database, many genes are absent, primarily because these genes have not yet been given a functional assignment (in the case of GO) or have not been allocated to an orthologous (ancestor-related) group (e.g., in the case of COG). In these instances, substitute values for the missing features are selected at random from the entire genome. Thus, missing values are replaced according to background frequencies, without bias toward the positive or negative sets.

Our positive and negative training sets are taken from ChIP-chip experiments [20, 36], TRANSFAC 6.0 Public [18], and a list from [37] curated by Young et al. from which we have excluded indirect evidence such as sequence analysis and expression correlation [38]. Only ChIP-chip interactions of *p*-value $\leq 10^{-3}$ are considered, as recommended by the authors [20]. The TRANSFAC and curated list represent a manually annotated set, which is later used separately during the comparison of SVM and PSSM performance. For the purposes of SVM, however, all manually curated and high-throughput sets are grouped together, making a total of 9,104 positive interactions. (The term *high-throughput* refers to the rapid processing of thousands of genes via ChIP-chip experiments.)

Negative sets pose a greater challenge because no defined negatives exist in the literature; however, because a particular TF regulates only a small fraction of the genome, a random choice of negatives seems acceptable.

In fact, our own unpublished work suggests that test cases with a few TFs show good classification performance with random negatives. Nevertheless, a more reliable set of negatives would be those showing no binding by experiment under some set of conditions. Along those lines, for each TF, we have chosen 175 genes with the highest p-values (generally >0.8) under all conditions tested in genomic ChIP-chip analyses [36]. Clearly, all experimental conditions have not been sampled, and this does not guarantee that our choices are truly never bound by the TF, but this choice of negatives maximizes our chances of selecting genes not regulated by the TF of interest.

All promoter sequences have been collected by using RSA tools, Ensembl, or the Broad Institute Fungal Genome Anatomy Project [39–41]. For yeast, promoters are defined as the 800 base pairs (bps) upstream of the coding sequence. The motif-conservation dataset required promoter regions from 17 other genomes. Those genomes, their sources, and the lengths of the promoter regions are listed in **Table 3**. Sequences are masked (i.e., replaced with a sequence of null characters) using the dust algorithm and the RepeatMasker software [42, 43] where it is appropriate to exclude low-complexity sequences and known repeat DNA from further analysis. PSSM scans are performed with the MotifScanner algorithm [44]. MotifScanner assumes a sequence model in which regulatory elements are distributed within a noisy background sequence [44]. The algorithm requires input of a background sequence model, which in this case is a transition matrix of a third-order Markov model generated from the masked upstream regions of each genome. MotifScanner requires only that one parameter be set by the user, namely the threshold score for accepting a motif as a binding site. Several thresholds have been tested, and the results we have used to create SVM kernels were obtained with a setting of 0.15 for the thresholds. This threshold has been found to provide a reasonable tradeoff between sensitivity and false prediction, making approximately 560 predictions per TF. Settings beyond 0.2 produce too many false hits to be useful. The PSSMs themselves are obtained from TRANSFAC 6.0 Public and from [45], and these PSSMs are a mix of experimentally derived motifs and those generated by motif-discovery procedures.

In addition, datasets using k-mers rather than PSSMs are generated using the fasta2matrix [52] program, which delineates all possible k-mers and counts the occurrence of each within a set of promoters. Gapped k-mers are detected using custom scripts written as MATLAB** m-files.

The expression data used include 1,011 microarray experiments complied by Ihmels and coworkers, and this data can be obtained with permission from the authors [53].

Table 3 Promoter regions.

Genome	*Promoter length*	*Source*
Human	clipped*	RSA tools [40]
Rat	clipped	RSA tools [40]
Fruit fly	clipped	RSA tools [40]
Anopheles mosquito	4,000 bp	Ensembl [46]
Worm	clipped	RSA tools [40]
S. pombe	800 bp	RSA tools [40]
S. cerevisiae	800 bp	RSA tools [40]
N. crassa	1,000 bp	Broad Institute [47]
M. grisea	1,000 bp	Broad Institute [48]
A. thaliana	clipped	RSA tools [40]
P. falciparum	clipped	RSA tools [40]
S. bayanus	clipped	Washington University [49]
S. mikatae	clipped	Washington University [49]
S. kluyveri	clipped	Washington University [49]
S. paradoxus	clipped	Broad Institute [50]
S. kudriavzevii	clipped	Washington University [49]
S. castellii	clipped	Washington University [49]
Mouse	clipped	Promoser [51]

*clipped: The promoter was truncated if it ran into an upstream coding sequence.

As mentioned above, 18 different data kernels are used to construct a classifier for each transcription factor. The datasets fall into several distinct groups. All classifier construction and validation was performed in MATLAB [54] using the SPIDER machine learning library [55].

Methods

First, each type of genomic data is evaluated independently for each transcription factor. Several kernel functions are tested, and parameters are optimized by a grid-selection technique. Each dataset is normalized so that all attributes describing the data have a mean of zero and a standard deviation of one. The Gene Ontology, phylogenetic profile, and TF-target correlation data are exceptions, and they are not normalized because their data is binary.

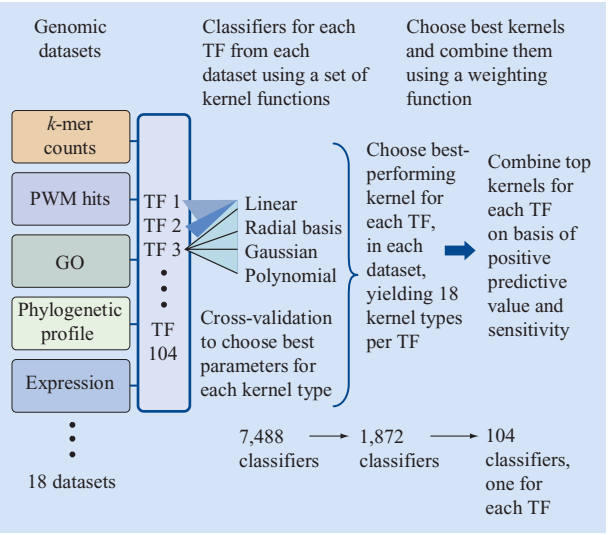

Figure 3

Flow diagram indicating the selection of a single classifier for each TF from several types of genomic data. A classifier is constructed for each individual TF for each genomic dataset, using each of four possible kernel functions (18 datasets × 104 TFs × 4 kernel functions = 7,488 total kernels from which SVM classifiers are built). For each dataset and each TF, the best-performing of the four kernel functions is selected, reducing the number of classifiers to 1,872 (18 datasets × 104 TFs). Finally, the datasets are combined on the basis of the F_1 score of their best-performing kernel so that there is only one classifier per TF.

A schematic representation of our method is shown in **Figure 3**. Briefly, for a particular TF, four classifiers are produced for each type of genomic data, each from a different kernel function (linear, RBF, Gaussian, and polynomial). In order to make an appropriate choice of the C parameter, a grid-selection technique is used to evaluate a range of choices. In the case of two parameter selections (e.g., when choosing the degree of the polynomial kernel), all possible combinations of parameter values within the pre-specified range are tested. A fivefold cross-validation is used to choose the best parameters on the basis of a Receiver Operating Characteristic (ROC) score. (The ROC score relates to the area under a Receiver Operating Characteristic curve that shows the utility of a classifier at various thresholds.)

Once parameters are chosen for each kernel type, the parameter-optimized classifiers are tested using a leave-one-out cross-validation procedure. As suggested, for each type of genomic data, there are four classifiers for a particular TF (one for each of the kernel functions). Of these four, we select the one with the best performance as measured by the F_1 statistic. Several common statistics, including accuracy, sensitivity, and specificity, can overstate the performance of a classifier depending on the relative size of the positive and negative training sets. The F_1 statistic is a more robust measure that is the harmonic mean between sensitivity (S) and positive predictive value (PPV):

$$F_1 = \frac{2 \times S \times PPV}{S + PPV} = \frac{2 \times TP}{2 \times TP + FP + FN}.$$

Each TF now has only one classifier for each type of genomic data (18 classifiers in all). Before weighting and combining kernels, each kernel matrix is normalized according to

$$\tilde{K}(x, y) = \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}}.$$

This normalization adjusts all points so that they lie on a unit hypersphere in the feature space. This ensures that no single kernel has matrix values that are comparatively larger or smaller than those of other kernels, which would bias the combination.

By using a scheme with weights equal to the F_1 of each classifier, the underlying 18 kernels are scaled and added into one unified kernel for the transcription factor. This kernel represents the integration of all types of genomic data. Three simple weighting schemes are compared. In all cases, the primary weight for a method is determined by computing its F_1 score ratio with that of the best-performing method. Our first weighting scheme simply multiplies all kernel matrices by their primary weights (i.e., F_1 ratios) and sums them. A second scheme squares the primary weights before multiplying. Our third scheme is the most nonlinear and requires us to compute the squared tangent of the primary weight.

Performance statistics for each TF, based on all combined datasets, were generated by a final leave-one-out cross-validation procedure on the combined kernel. In this way, accuracy measurements are made for each TF–target classifier.

PSSM comparison

Using the same positive and negative sets as are used for the SVM procedure, PSSMs can make predictions at various score thresholds to serve as a comparison to predictions made by SVMs. The data in **Figure 4** represent a parameter setting of only 0.1 in MotifScanner. Low parameter values retain the best matches, whereas values near 1 allow very "loose hits"; that is, the use of values near 1 leads to the retention of more false matches. Other choices of threshold do not appear to improve performance. Loosening the threshold begins to dramatically increase false-positive predictions beyond a parameter setting of 0.2. By making detection more "strict" (i.e., less likely to yield false hits), false predictions are reduced along with sensitivity. Because the matrices for the 104 transcription factors are partly

Table 4 Performance results of combined classifier and random datasets.

	Combined methods	*Random* k-*mers*	*Random 10% of all datasets*	*Random normal data*
Accuracy	0.88	0.67	0.58	0.58
Sensitivity	0.73	0.45	0.62	0.61
PPV	0.88	0.50	0.41	0.41
F1	0.80	0.48	0.50	0.50

experimentally determined and partly computationally generated, the TRANSFAC PSSMs for 17 TFs are evaluated next to determine whether the experimental matrices by themselves outperform SVM for target identification. Finally, because a large number of positive targets have been taken from high-throughput ChIP-chip experiments, the TRANSFAC PSSMs are tested again on only the portion of the positives obtained from manually annotated sources.

Results and discussion

Using the classification procedure described in the previous sections, we have been able to accurately classify the known targets of many transcription factors for the yeast *S. cerevisiae*. Overall, the best single method achieves a sensitivity of 71% and a positive predictive value of 0.82. These performance measures provide a summary for all 104 classifiers. For example, there are 9,104 known positives for all TFs. A sensitivity of 71% indicates that, taking into account all 104 classifiers, we recover 71% of the known data (i.e., known TF–target interactions). This means that classifiers for some TFs have much higher sensitivities or PPVs, while other classifiers perform no better than randomly. Many individual methods perform well, but the best classification is made with *k*-mer counts allowing one mismatch per *k*-mer (with mismatches given a count of 0.1). Our results show that by combining datasets we increase sensitivity incrementally over the use of only the best single dataset, and also produce a small improvement in positive predictive value. This indicates that methods that combine data sources are useful in this case because they remove some false-positive classifications [35].

To prevent an overly optimistic evaluation of our performance, we generated three random datasets and trained TF classifiers on them as if they were actual data. Comparison with random controls better frames the practical performance of our method. The first random set consists of randomly permuted *k*-mer count data. The second is composed of a randomly selected 10% of each real dataset (also permuted). The third is a dataset composed of normally distributed random numbers in the range 0 to 1. The comparison of these results is shown in **Table 4**.

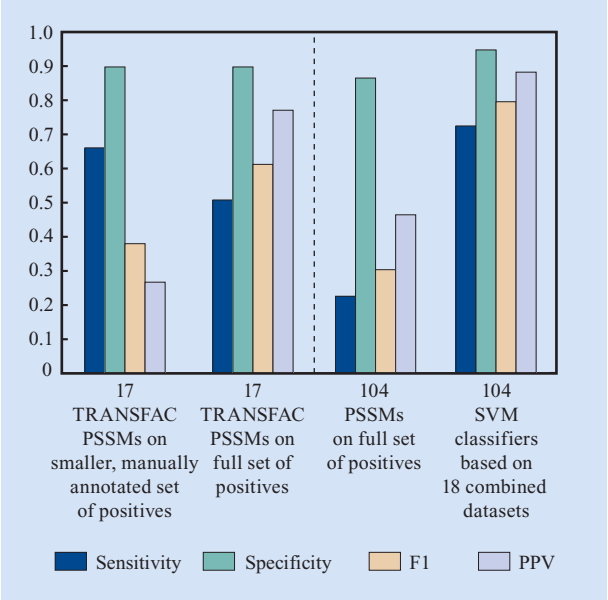

Figure 4

Comparison of SVM and PSSM scans, indicating that SVM classifiers outperform PSSMs. The same negative sets were used for all scans. The *y*-axis represents values of sensitivity, specificity, PPV, and F1 in the range of 0 to 1.

Clearly, the performance is much better than random, but results do not clearly indicate whether applying our classifiers to the entire genome would yield truly reliable predictions without further processing. A simple classification of all potential targets with our 104 classifiers returns, on average, approximately 800 new targets for each TF. This suggests that in order to find a set of truly reliable predictions genome-wide, postprocessing of our results is needed. Indeed, in other work we have applied Platt's method [56] to assign "posterior probabilities" to our predictions, allowing the selection of only the most significant targets [35]. The precise meaning of the term *posterior probabilities* is clarified in [35]. Using these probabilistic SVMs, the classifiers for each TF were applied to identify potential targets genome-wide in order to expand the binding repertoire of each factor. This results in predictions of new regulatory roles for some TFs and the identification of possible new regulatory structures such as feed-forward loops in metabolic pathways [35].

Many reports in the literature indicate that as many as 50% [57] to 60% [58] of the targets produced by ChIP-chip are not biologically functional. Our ability to correctly classify large amounts of high-throughput data indicates that there is relevant biological information that identifies

ChIP-chip positives from other genes. One should also note that ChIP-chip experiments may be highly accurate in detecting binding of a TF even if that binding serves no biological function. This may be interpreted as a false positive from a functional perspective, but not so from a binding perspective. Our experiments may accurately classify binding targets as identified by ChIP-chip even if those targets show no change in expression as a result of binding.

In other work, we search for evidence that the predictions based on various classifiers make biological sense [35]. To do this, we examine individual datasets and extract the attributes that contribute most to the classifier of a transcription factor. The **w** vector described in previous paragraphs can be used in this way to identify the features, in any particular dataset, that are most important for classification. Features having large **w** components correspond to dimensions in the feature space where positives and negatives are more definitively separated. Thus, by examining a single dataset such as one that includes *k*-mer counts, it is possible to determine the *k*-mer(s) most responsible for the differences between positives and negatives. Our results on the *k*-mer count dataset have shown that the many *k*-mers having large **w** values are in fact elements of the known transcription factor binding site as taken from the Saccharomyces Genome Database (SGD) [35].

To better judge the performance of new methods, it is sometimes useful to compare them with standard PSSM scans for their ability to identify targets. Carefully constructed variants of PSSMs, which take into account conservation of sites between multiple species or dependencies between nucleotides, offer excellent performance, but often there is insufficient data to construct such detailed models. In TRANSFAC version 6, only 17 available binding site matrices exist for yeast. Many of the remaining PSSMs used in this study have been created using motif discovery methods on high-throughput datasets [20]. The purpose of our comparison with PSSMs is to illustrate that some of the commonly used site matrices perform worse than a classification scheme built on an integrated dataset.

Overall, the SVM performs better than a simple weight-matrix scan. Figure 4 shows such a comparison as a function of sensitivity, specificity, positive predictive value, and the F_1 statistic. The far-left grouping of data uses the TRANSFAC PSSMs for 17 TFs on just the manually curated positives (with same negatives as all other analyses) from TRANSFAC and literature sources. The second grouping from the left uses the same TRANSFAC PSSMs as the first grouping, but this time with the same high-throughput positive sets used in the SVM classification. The third grouping is a result from scans using PSSMs for all 104 TFs on the positive and negative sets on which the SVMs were trained. Finally, the far-right grouping restates the performance of the SVMs with 18 combined datasets on the full set of positives. The SVM classifiers outperform PSSMs, even when the matrices are from a curated set such as TRANSFAC. Although the PSSMs perform well, they suffer from a large number of false-positive predictions. Figure 4 shows data for only one threshold of PSSM scan, but altering the threshold does not make PSSMs more accurate than SVMs (see the Methods section). It is worth noting, however, that the site matrices from TRANSFAC offer much better performance than the matrices generated by motif-discovery procedures. Support vector machine classifiers offer a reasonable balance between sensitivity and false prediction. Alternatives to SVMs, such as Bayesian networks and neural networks, may offer similar performance, but SVMs have an advantage because they permit different types of high-dimensional data to be easily combined.

Concluding remarks

In conclusion, support vector machines can accurately classify and predict transcription factor binding sites using a wide range of genomic data types. Combining various information sources reduces false positives and increases sensitivity. On the basis of *k*-mer data, SVMs appear to be identifying appropriate features for classification. Finally, the flexibility of this approach allows easy inclusion of new types of genomic data. Our future work involves the development of sophisticated dimension-reduction techniques to discover biologically significant features in different datasets on the basis of classifier performance. As always with high-dimensional datasets, the risk of over-fitting can restrict the wide application of a classification tool. (The term *over-fitting* is considered to be synonymous with *overtraining*, which indicates that a classifier is very accurate for a training set but less accurate for independent test sets.) Although the maximal margin of SVMs is resistant to over-fitting, the resistance can be enhanced by selecting the best features for classifier construction. In future work, we plan to test several feature-reduction methods such as Fisher's Linear Discriminant and SVM–RFE (Recursive Feature Elimination). A reduction in the feature set would also allow a comparison with other classification systems, such as Bayesian networks or KNN classifiers, which are difficult to train on very large sets of features.

Additionally, new datasets can be included that leverage information about DNA structural features. Information of this type could include promoter melting-temperature profiles, bend and curve features of promoters [59], or DNA accessibility predictions based on patterns of hydroxyl radical cleavage [60, 61]. Furthermore, it may be possible to capture more meaningful information from *k*-mer counts by

additionally measuring the likelihood that a certain k-mer occurs by chance in a gene's promoter, thus attaching a p-value to all k-mers in each promoter region. Support-vector machines show promise as a means to analyze regulatory relationships and will be increasingly useful for the analysis of mammalian genomes as more genomic data becomes available.

**Trademark, service mark, or registered trademark of BIOBASE GmbH, The MathWorks, Inc., or Microsoft Corporation in the United States, other countries, or both.

References

1. E. M. Conlon, X. S. Liu, J. D. Lieb, and J. S. Liu, "Integrating Regulatory Motif Discovery and Genome-Wide Expression Analysis," *Proc. Natl. Acad. Sci.* **100**, No. 6, 3339–3344 (2003).
2. S. Keles, M. J. van der Laan, and C. Vulpe, "Regulatory Motif Finding by Logic Regression," *Bioinformatics* **20**, No. 16, 2799–2811 (2004).
3. W. Wang, J. M. Cherry, D. Botstein, and H. Li, "A Systematic Approach to Reconstructing Transcription Networks in Saccharomyces scerevisiae," *Proc. Natl. Acad. Sci.* **99**, No. 26, 16893–16898 (2002).
4. H. Bussemaker, H. Li, and E. Siggia, "Regulatory Element Detection Using Correlation with Expression," *Nature Genetics* **27**, No. 2, 167–171 (2001).
5. K. Birnbaum, P. N. Benfey, and D. E. Shasha, "cis Element/Transcription Factor Analysis (cis/TF): A Method for Discovering Transcription Factor/cis Element Relationships," *Genome Res.* **11**, No. 9, 1567–1573 (2001).
6. Z. Zhu, Y. Pilpel, and G. Church, "Computational Identification of Transcription Factor Binding Sites via a Transcription-Factor-Centric-Clustering (TFCC) Algorithm," *J. Molec. Biol.* **318**, No. 2, 71–81 (2002).
7. M. Pritsker, Y.-C. Liu, M. A. Beer, and S. Tavazoie, "Whole-Genome Discovery of Transcription Factor Binding Sites by Network-Level Conservation," *Genome Res.* **14**, No. 1, 99–108 (2004).
8. S. Elemento and S. Tavazoie, "Fast and Systematic Genome-Wide Discovery of Conserved Regulatory Elements Using a Non-Alignment Based Approach," *Genome Biol.* **6**, No. 2, R18 (2005).
9. M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu, "Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites," *Nature Biotechnol.* **23**, No. 1, 137–144 (2005).
10. G. D. Stormo, "DNA Binding Sites: Representation and Discovery," *Bioinformatics* **16**, No. 1, 16–23 (2000).
11. C. T. Workman and G. D. Stormo, "ANN-Spec: A Method for Discovering Transcription Factor Binding Sites with Improved Specificity," *Proceedings of the Pacific Symposium on Biocomputing,* 2000, pp. 467–478.
12. T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht, "Information Content of Binding Sites on Nucleotide Sequences," *J. Molec. Biol.* **188**, No. 3, 415–431 (1986).
13. T. Schneider and R. Stephens, "Sequence Logos: A New Way to Display Consensus Sequences," *Nucl. Acids Res.* **18**, No. 20, 6097–6100 (1990).
14. M. C. Frith, M. C. Li, and Z. Weng, "Cluster-Buster: Finding Dense Clusters of Motifs in DNA Sequences," *Nucl. Acids Res.* **31**, No. 13, 3666–3668 (2003).
15. B. P. Berman, Y. Nibu, B. D. Pfeiffer, P. Tomancak, S. E. Celniker, M. Levine, G. M. Rubin, and M. B. Eisen, "Exploiting Transcription Factor Binding Site Clustering to Identify Cis-Regulatory Modules Involved in Pattern Formation in the Drosophila Genome," *Proc. Natl. Acad. Sci.* **99**, No. 2, 757–762 (2002).
16. D. Dinakarpandian, V. Raheja, S. Mehta, E. Schuetz, and P. Rogan, "Tandem Machine Learning for the Identification of Genes Regulated by Transcription Factors," *BMC Bioinformatics* **6**, No. 1, 204 (2005).
17. M. Rebeiz, N. L. Reeves, and J. W. Posakony, "SCORE: A Computational Approach to the Identification of Cis-Regulatory Modules and Target Genes in Whole-Genome Sequence Data," *Proc. Natl. Acad. Sci.* **99**, No. 15, 9888–9893 (2002).
18. V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender, "TRANSFAC® and Its Module TRANSCompel® Transcriptional Gene Regulation in Eukaryotes," *Nucl. Acids Res.* **34**, No. 1, D108–D110 (2006).
19. C. Lemer, E. Antezana, F. Couche, F. Fays, X. Santolaria, R. S. Janky, Y. Deville, J. Richelle, and S. J. Wodak, "The aMAZE LightBench: A Web Interface to a Relational Database of Cellular Processes," *Nucl. Acids Res.* **32**, D443–D448 (2004).
20. C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J.-B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young, "Transcriptional Regulatory Code of a Eukaryotic Genome," *Nature* **431**, No. 7004, 99–104 (2004).
21. B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young, "Genome-Wide Location and Function of DNA Binding Proteins," *Science* **290**, No. 5500, 2306–2309 (2000).
22. J. Qian, J. Lin, N. M. Luscombe, H. Yu, and M. Gerstein, "Prediction of Regulatory Networks: Genome-Wide Identification of Transcription Factor Targets from Gene Expression Data," *Bioinformatics* **19**, No. 15, 1917–1926 (2003).
23. M. A. Beer and S. Tavazoie, "Predicting Gene Expression from Sequence," *Cell* **117**, No. 2, 185–198 (2004).
24. D. Holloway, M. Kon, and C. DeLisi, "Integrating Genomic Data to Predict Transcription Factor Binding," *Proc. Workshop Genome Informatics* **16**, No. 1, 83–94 (2005).
25. T. Jaakola, M. Diekhans, and D. Haussler, "Using the Fisher Kernel Method to Detect Remote Protein Homologies," *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, August 6–10, 1999, pp. 149–158.
26. S. Hua and Z. Sun, "A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach," *J. Molec. Biol.* **308**, No. 2, 397–407 (2001).
27. S. Hua and Z. Sun, "Support Vector Machine Approach for Protein Subcellular Localization Prediction," *Bioinformatics* **18**, No. 8, 721–728 (2001).
28. M. Wang, J. Yang, and K.-C. Chou, "Using String Kernel to Predict Signal Peptide Cleavage Site Based on Subsite Coupling Model," *Amino Acids* **28**, No. 4, 395–402 (2005).
29. T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data," *Bioinformatics* **16**, No. 10, 906–914 (2000).
30. P. Pavlidis and W. S. Noble, "Gene Functional Classification from Heterogeneous Data," *RECOMB Conference Proceedings,* 2001, pp. 249–255.
31. A. Zien, G. Ratsch, S. Mika, B. Scholkopf, T. Lengauer, and K.-R. Muller, "Engineering Support Vector Machine Kernels

That Recognize Translation Initiation Sites," *Bioinformatics* **16**, No. 9, 799–807 (2000).
32. G. Lanckriet, N. Cristianini, M. Jordan, and W. S. Noble, "A Statistical Framework for Genomic Data Fusion," *Bioinformatics* **20**, No. 16, 2626–2635 (2004).
33. B. Scholkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
34. P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley Publishing Co., Boston, MA, 2005.
35. D. Holloway, M. Kon, and C. DeLisi, "Machine Learning and Data Combination for Regulatory Pathway Prediction," *Synthetic & Syst. Biol.* (2006), submitted.
36. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional Regulatory Networks in Saccharomyces cerevisiae," *Science* **298**, No. 5594, 799–804 (2002).
37. P. Hodges, A. McKee, B. Davis, W. Payne, and J. Garrels, "The Yeast Proteome Database (YPD): A Model for the Organization and Presentation of Genome-Wide Functional Data," *Nucl. Acids Res.* **27**, No. 1, 69–73 (1999).
38. R. Young, "Transcriptional Regulatory Network"; see *http://staffa.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=evidence*.
39. M. Kellis et al.,"Yeast Comparative Genomics"; see *http://www.broad.mit.edu/annotation/fungi/comp_yeasts/* (2003).
40. J. van Helden, "Regulatory Sequence Analysis Tools," *Nucl. Acids Res.* **31**, No. 13, 3593–3596 (2003).
41. E. Birney, T. D. Andrews, P. Bevan, M. Caccamo, Y. Chen, L. Clarke, G. Coates, J. Cuff, V. Curwen, T. Cutts, T. Down, E. Eyras, X. M. Fernandez-Suarez, P. Gane, B. Gibbins, J. Gilbert, M. Hammond, H.-R. Hotz, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, S. Keenan, H. Lehvaslaiho, G. McVicker, C. Melsopp, P. Meidl, E. Mongin, R. Pettett, S. Potter, G. Proctor, M. Rae, S. Searle, G. Slater, D. Smedley, J. Smith, W. Spooner, A. Stabenau, J. Stalker, R. Storey, A. Ureta-Vidal, K. C. Woodwark, G. Cameron, R. Durbin, A. Cox, T. Hubbard, and M. Clamp, "An Overview of Ensembl," *Genome Res.* **14**, No. 5, 925–928 (2004).
42. R. L. Tatusov and D. J. Lipman, "National Center for Biotechnology Information, NCBI Toolkit"; see *http://www.ncbi.nlm.nih.gov/*.
43. A. Smit, R. Hubley, and P. Green, Institute for Systems Biology, "Repeatmasker Open 3.0"; see *http://repeatmasker.org*.
44. S. Aerts, G. Thijs, B. Coessens, M. Staes, Y. Moreau, and B. De Moor, "Toucan: Deciphering the Cis-Regulatory Logic of Coregulated Genes," *Nucl. Acids Res.* **31**, No. 6, 1753–1764 (2003).
45. C. Harbison, E. Fraenkel, and R. Young, "Matrices for Motifs"; see *http://jura.wi.mit.edu/fraenkel/download/release_v24/final_set/Final_InTableS2_v24.motifs*.
46. E. Birney, D. Andrews, M. Caccamo, Y. Chen, L. Clarke, G. Coates, T. Cox, F. Cunningham, V. Curwen, T. Cutts, T. Down, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, S. Graf, M. Hammond, J. Herrero, K. Howe, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, F. Kokocinski, E. Kulesha, D. London, I. Longden, C. Melsopp, P. Meidl, B. Overduin, A. Parker, G. Proctor, A. Prlic, M. Rae, D. Rios, S. Redmond, M. Schuster, I. Sealy, S. Searle, J. Severin, G. Slater, D. Smedley, J. Smith, A. Stabenau, J. Stalker, S. Trevanion, A. Ureta-Vidal, J. Vogel, S. White, C. Woodwark, and T. J. P. Hubbard, "Ensembl 2006," *Nucl. Acids Res.* **34**, No. 1, D556–D561 (2006).
47. J. E. Galagan, S. E. Calvo, K. A. Borkovich, E. U. Selker, N. D. Read, D. Jaffe, W. FitzHugh, L.-J. Ma, S. Smirnov, S. Purcell, B. Rehman, T. Elkins, R. Engels, S. Wang, C. B. Nielsen, J. Butler, M. Endrizzi, D. Qui, P. Ianakiev, D. Bell-Pedersen, M. A. Nelson, M. Werner-Washburne, C. P. Selitrennikoff, J. A. Kinsey, E. L. Braun, A. Zelter, U. Schulte, G. O. Kothe, G. Jedd, W. Mewes, C. Staben, E. Marcotte, D. Greenberg, A. Roy, K. Foley, J. Naylor, N. Stange-Thomann, R. Barrett, S. Gnerre, M. Kamal, M. Kamvysselis, E. Mauceli, C. Bielke, S. Rudd, D. Frishman, S. Krystofova, C. Rasmussen, R. L. Metzenberg, D. D. Perkins, S. Kroken, C. Cogoni, G. Macino, D. Catcheside, W. Li, R. J. Pratt, S. A. Osmani, C. P. C. DeSouza, L. Glass, M. J. Orbach, J. A. Berglund, R. Voelker, O. Yarden, M. Plamann, S. Seiler, J. Dunlap, A. Radford, R. Aramayo, D. O. Natvig, L. A. Alex, G. Mannhaupt, D. J. Ebbole, M. Freitag, I. Paulsen, M. S. Sachs, E. S. Lander, C. Nusbaum, and B. Birren, "The Genome Sequence of the Filamentous Fungus *Neurospora crassa*," *Nature* **422**, No. 6934, 859–868 (2003).
48. R. Dean, "Fungal Genomics Laboratory at North Carolina State University, Broad Institute of MIT and Harvard"; see *http://www.fungalgenomics.ncsu.edu* and *http://www.broad.mit.edu*.
49. P. Cliften, P. Sudarsanam, A. Desikan, L. Fulton, B. Fulton, J. Majors, R. Waterston, B. A. Cohen, and M. Johnston, "Finding Functional Features in Saccharomyces Genomes by Phylogenetic Footprinting," *Science* **301**, No. 5629, 71–76 (2003).
50. M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander, "Sequencing and Comparison of Yeast Species to Identify Genes and Regulatory Elements," *Nature* **423**, No. 6037, 241–254 (2003).
51. A. Halees, D. Leyfer, and Z. Weng, "Promoser: A Larger-Scale Mammalian Promoter and Transcription Start Site Identification Service," *Nucl. Acids Res.* **31**, No. 13, 3554–3559 (2003).
52. P. Pavlidis, I. Wapinski, and W. S. Noble, "Support Vector Machine Classification on the Web," *Bioinformatics* **20**, No. 4, 586–587 (2004).
53. J. Ihmels, S. Bergman, and N. Barkai, "Naama Barkai Group"; see *http://barkai-serv.weizmann.ac.il/GroupPage/*.
54. The Mathworks, "MATLAB: MATrix LABoratory"; see *http://www.mathworks.com/*.
55. J. Weston, A. Elisseeff, G. Bakir, and F. Sinz, "SPIDER: Object Oriented Machine Learning Library"; see *http://www.kyb.tuebingen.mpg.de/bs/people/spider/*.
56. J. C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*, P. Bartlett, B. Schölkopf, D. Schuurmans, and A. Smola, Eds., MIT Press, Cambridge, MA, 2000.
57. N. Simonis, S. J. Wodak, G. N. Cohen, and J. van Helden, "Combining Pattern Discovery and Discriminant Analysis to Predict Gene Co-Regulation," *Bioinformatics* **20**, No. 15, 2370–2379 (2004).
58. F. Gao, B. Foat, and H. Bussemaker, "Defining Transcriptional Networks Through Integrative Modeling of mRNA Expression and Transcription Factor Binding Data," *BMC Bioinformatics* **5**, No.1, 31 (2004).
59. D. Goodsell and R. Dickerson, "Bending and Curvature Calculations in B-DNA," *Nucl. Acids Res.* **22**, No. 24, 5497–5503 (1994).
60. S. Parker, J. Greenbaum, G. Benson, and T. D. Tullius, "Structure-Based DNA Sequence Alignment," poster presented at the 5th International Workshop in Bioinformatics and Systems Biology, Berlin, Germany, August 2005.
61. B. Balasubramanian, W. K. Pogozelski, and T. D. Tullius, "DNA Strand Breaking by the Hydroxyl Radical Is Governed by the Accessible Surface Areas of the Hydrogen Atoms of the DNA Backbone," *Proc. Natl. Acad. Sci.* **95**, No. 17, 9738–9743 (1998).

Received October 3, 2005; accepted for publication December 21, 2005; Internet publication June 27, 2006

Dustin T. Holloway *Department of Molecular Biology, Cell Biology, and Biochemistry, Boston University, Boston, Massachusetts 02215 (dth128@bu.edu)*. Mr. Holloway received his bachelor's degree in microbiology with a minor in biochemistry from Pennsylvania State University in 2002. He is currently a Dean's Fellow in the Molecular Biology, Cell Biology, and Biochemistry (MCBB) Department and a Ph.D. candidate in Dr. Charles DeLisi's laboratory.

Mark A. Kon *Department of Mathematics and Statistics, Boston University, Boston, Massachusetts 02215 (mkon@bu.edu)*. Dr. Kon is a professor of mathematics and statistics at Boston University; he is affiliated with the Department of Cognitive and Neural Systems and the Bioinformatics Graduate Program. He has also served as departmental director of graduate studies at Boston University and is on the editorial board of *Neural Networks*.

Charles DeLisi *Department of Bioinformatics and Systems Biology, Boston University, Boston, Massachusetts 02215 (delisi@bu.edu)*. Dr. DeLisi is Arthur G. B. Metcalf Professor of Science and Engineering at Boston University, where he served as Dean of the College of Engineering from 1990 to 2000. He is also a former director of the Department of Energy Health and Environmental Research Programs and a current Fellow of the AAAS and the American Institute of Medical and Biological Engineers.