SVMotif: A Machine Learning Motif Algorithm

Mark Kon*, Yue Fan
Mathematics and Statistics
Boston University
Boston, MA 02215
mkon@bu.edu
yue@bu.edu

Dustin Holloway
Molecular Biology, Cell Biology, and Biochemistry
Boston University
Boston, MA 02215
dth128@bu.edu

Charles DeLisi
Bioinformatics and Systems Biology
Boston University
Boston, MA 02215
delisi@bu.edu

Abstract

We describe SVMotif, a support vector machine-based learning algorithm for identification of cellular DNA transcription factor (TF) motifs extrapolated from known TF-gene interactions. An important aspect of this procedure is its ability to utilize negative target information (examples of likely non-targets) as well as positive information. Applications involve situations where clusters of genes are distinguished in experiments with known transcription factors without known binding locations. We apply this to yeast TF data with target identifications from ChIP-chip and other sources, and compare performance with Gibbs sampling methods such as BioProspector. We verify that in yeast this method implies well-defined and cross-validated statistical correlations between TF binding and secondary motifs whose binding properties (either with the primary TF or other possible promoters) are not certain, and discuss some implications of this. SVMotif can be a useful standalone method or a complement to existing techniques, and it will be made publicly available.

1. Introduction

We will describe a machine-based method which can augment and improve on existing ones for discovering motifs from experimental information on transcription factor (TF)-gene interactions. This is a high dimensional problem which has been successfully approached using Gibbs sampling methods (e.g., [30], [21]) among others, and is now also being studied using machine learning methods ([36], [17]). We will illustrate our machine learning approach and compare it to Gibbs sampling for an important data set - TF binding information for the budding yeast *S. cerevisiae* (baker's yeast).

There are many DNA components near a given gene in a eukaryotic cell which influence its transcription and resulting activation for protein production (expression). These DNA regions can bind activators, co-activators, and inhibitors. Often these act to reinforce or inhibit each other. In general they can act combinatorially, determining the gene's expression depending on circumstances. Primary among such binding proteins are TF's, the main activators of gene transcription.

The DNA binding components are largely determined by binding site motifs, characteristic signatures of length 5-15 DNA base pairs (bp) which bind to transcription factors (TF's). The number of copies of a given TF's binding motif in a gene's promoter region (a region known to contain most TF binding sites) is a strong predictor of the gene's response to the TF, and thus an indicator for the gene's regulation and function. For this reason the characterization of these DNA motifs has become an important biological problem.

The determination of a given TF's binding motifs often follows from identification of common DNA patterns in promoter regions (regions known to contain most TF binding sites) adjacent to genes which are known experimentally to bind the TF, or shown to express their RNA or protein in the presence of the TF in experiments. Methodologies for obtaining this information include microarray experiment correlations ([18], [25], [3]), phylogeny [2], and gene ontology [6].

A common method has been based on Gibbs sampling to optimize alignment among such gene clusters, in order to determine the most common motif pattern there ([30], [21], [27]). In BioProspector, for example, an optimal alignment is achieved by initiating an arbitrary alignment among positive promoter regions, and adjusting this alignment one promoter at a time until an optimal one is found. The Gibbs sampling is done with a probabilistic simulated annealing algorithm in order to avoid local minima.

These methods have more recently been augmented by machine learning approaches. In [18], [19] for example, boosting methods are used to accumulate weak rules related to presence of given k-mers (consecutive strings of length k appearing in the gene's promoter region) and given regulators (TF's) as predictors of gene expression. This is done in experiments involving large numbers of regulators and various gene expression outcomes. From these statistical relationships between k-mer presence, regulator presence, and gene expression, there is produced an 'important set' of k-mers statistically associated with expression via a regulator. These are then agglomerated into position weight matrices (PWM's) representing probabilities of DNA bases in given locations of likely binding site motifs. In [36] it is shown how incorporating hierarchical kernel information into analysis of promoter regions can result in accurate predictions of binding between TF's and gene promoters.

We present here a machine learning method, SVMotif, which finds motifs by statistical association of k-mers with known interactions of TF's with genes, learning them from these examples. In its attempt to optimize based on similarities between binding promoters, this method acts similarly to BioProspector and other methods; however, it also has the advantage of using negative (i.e., genes whose promoters are expected not to bind) as well as positive examples in learning a TF binding rule (see below for some other approaches which also use negative information). We expect that it will also be possible to find binding motifs using other machine learning techniques (e.g., random forests, least angle regression [7]) which determine important variables out of a large class; such additional approaches will be examined in later work.

This method uses k-mer importance determination from the largest components of a modification of the SVM gradient vector $\mathbf{w}$ to statistically determine those k-mers which are most associated with a TF. SVM is a kernel-based method [34], [35], [4], [31] which incorporates prior information via mappings Φ of genomic feature vectors $\mathbf{x}$ which effectively change the geometry of the feature space. SVM have been used in genomics in a number of different ways [26], [20], [32].

SVMotif extracts motifs from data sets involving promoters of genes known to bind a given TF (with a margin of error which can be allowed in input data; see below); it will be made publicly available. As mentioned, a general advantage of learning methods is use of negative and positive examples, which is not available in Gibbs sampling. Negative examples can be chosen in a few ways, including random selection of promoter regions from the general population of genes (since being negative is a statistical likelihood for a random promoter), which can produce an error margin that must be statistically acceptable. Negatives can also be found from members of the gene population with highest p values for binding the promoter of a given gene g, as in the case of chromatin immunoprecipitation chip (ChIP-chip) data [8], [29]) - we use such choices of negatives here.

In cases where only examples of (positive) targets are available (though likely non-targets could be generated from the background gene distribution as above), negatives can be artificially generated from permutation of positive examples, with conservation of either zeroth or higher order Markov properties. This has been done by us for the data described below with 0th order Markov preservation (i.e., only of overall nucleotide probabilities), and has been shown to decrease prediction accuracy only by about 10% to 15%.

The table below shows differences in sensitivity for a few typical TF's, here chosen from the subset of YeastGenome ([14], http://www.yeastgenome.org) transcription factors which also have motifs listed in Transfac [24], a curated database.

Name	#pos	YeastGenome (Transfac)	BioProspector	SVMotif
YBR049C	186	CGGGTRR TTACCCG	CGGGTAA TACCCGG	AAGAAGARG CYTCTTCTT GGCGGGTAAC
YDL020C	134	GGTGGCAAA	GGCGGGTAA GTTTCCCCG GTTTCCCCG	CCGGTGGCRG TSGCCACCSG AAGAAGAGG
YDL056W	207	ACGCGT	TACATA GCGACT GGTTGG	ARACGCGTYT GYYTTCTTS SAAGAARRC
YDL106C	69	SGTGCGSYGYG	ATCCTCGAGTT GACTCACAATC GCACTTACAAC	CSCCACGTGGG CCGCTGCAGCG CCCGGG

Table 1: A sample of yeast transcription factors which have been analyzed in this paper . # pos represents the number of positive examples for TF. The standard motifs (middle column) appear in Transfac [20] but are taken from YeastGenome [14] (which takes biological input from Transfac [24]). Motifs to the right are in order of priority.

Other related work: This approach identifies statistical properties of k-mers occurring more frequently in one group of genes than another. As mentioned above, boosting [18], [19] and support vector machine (SVM) approaches [36] have been used to demonstrate machine learning as a viable methodology in the area of determining binding genes and binding sites of TF's. In particular kernel methods have been used successfully in the work of Noble and Vert [36] to determine genes-TF associations.

The central aspect here is learning-based ranking of k-mers from correlation with promoters known to bind a given TF t. Among other places, this is also done in [36], which uses kernels on feature vectors of 5-mer counts. Let

$\mathbf{x}(g)$ be the sequence of bases in the promoter $\mathbf{x}$ of g. The feature vector $\Phi(\mathbf{x}) \equiv \Phi^{\text{spect}}(\mathbf{x})$ of the promoter region is a vector of length 4^5, with each position $\Phi_i(\mathbf{x})$ representing the count of the i^{th} 5-mer in the indexing sequence. An interesting addition in [36] is use of feature vectors taking phylogenetic conservation into account. They consider a given upstream region $\mathbf{x} \in \mathcal{A}^n$ of *S. cerevisiae*, and an alignment $\mathbf{c} \in (\mathcal{A}^5)^n$, consisting of an aligned array of five (matching) upstream regions of length n from 5 related yeast species, to determine functional regions. Above $\mathcal{A}$ is the alphabet {A, C, G, T}.

There have been some other methods using negative examples (likely nonbinding genes for a TF) for determining binding motifs. An example is the Ann-Spec algorithm [37]. Some boosting motif finding methods use negative information [13] as well.

2. SVM feature space approach

We assume a fixed TF t, and a dataset from experiments which identify promoter regions of genes activated by t (in the present case these are ChIP-chip experiments [28], [8]). Our methods can be adapted to other data sets, for example those in [18] which map activation of a gene g as a function of TF presence together with presence of given motifs in the promoter region of g.

We have for the promoter sequence $\mathbf{x}(g)$ of g a classifier $y = \pm 1$ indicating (with allowed error) whether or not t binds the promoter of g.

The SVM gives a function on promoters, defined by

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \equiv \mathbf{w} \cdot \mathbf{x} + b,$$

with $K(\mathbf{x}, \mathbf{y})$ a string kernel ([17]) given by

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}), \tag{1}$$

with Φ a feature map taking $\mathbf{x}$ into the k-mer feature vector above. Software implementing this algorithm includes:

• SVMLight: http://svmlight.joachims.org
•SVMTorch: http://www.idiap.ch/learning/SVMTorch.html
• LIBSVM: http://wws.csie.ntu.edu.tw/~cjlin/libsvm

A Matlab package which implements most SVM algorithms with a C-based back end is SPIDER:
http://www.kyb.mpg.de/bs/people/spider/whatisit.html
The latter implementation is used here.

We illustrate the algorithm (again for t fixed), given an example data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of promoter regions $\mathbf{x}_i$ and outcomes y_i (binding/no binding). $\mathcal{D}$ is allowed to have large errors (as are sometimes assumed to exist e.g. in ChIP-chip experiments [9], [33]), as long as statistical patterns are not masked by very small n.

For g a gene in *S. cerevisiae*, $\mathbf{x}(g)$ is the FASTA sequence for its upstream region, up to 800 bp long. For fixed t an SVM with a linear string kernel $K(\mathbf{x}, \mathbf{y})$ defined in (1) takes data $\mathcal{D}$ and forms a discriminator $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ which separates positive examples ($(\mathbf{x}_i, y_i)$ for which $y_i = 1$) from negative ones. Using the R-SVM feature reduction procedure [38], an importance vector $\tilde{\mathbf{w}} = \mathbf{w} \times (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$ is formed. Here, $\boldsymbol{\mu}_+$ represents the center (vector mean) of the positive examples in the feature space F, while $\boldsymbol{\mu}_-$ represents the center of the negative examples. The vector $\mathbf{a} \times \mathbf{b}$ represents a componentwise multiplication of $\mathbf{a}$ and $\mathbf{b}$. From this a permutation π of indices of the vector $\tilde{\mathbf{w}}$ is formed so the permuted vector $\tilde{\mathbf{w}} \circ \pi \equiv \tilde{\mathbf{w}}_\pi$ is arranged from largest to smallest component. The R-SVM program takes the components of $\tilde{\mathbf{w}}_\pi$ and reduces them to 150. This feature selection is repeated 20 times with different choices of presumed negatives (in this case genes with high p values in ChIP-chip experiments) out of about 600 negative examples available. In each SVM run the numbers of positives and negatives are chosen to be equal. From the top 600 k-mers obtained in these 20 runs, the best 50 are chosen using a single iteration of R-SVM. For typical transcription factors here, there are approximately 50-300 positive examples (i.e., genes binding t), and a larger number (e.g. 600) of negative examples, from which different samples are taken in multiple runs to match numbers of positives.

Though division into training and test sets is not necessary in the motif finding procedure above, it is still valuable to know how good the SVM algorithm is in predicting whether a gene will bind the given TF, i.e., how good $\mathbf{w}$ is at segregating the positive from the negative examples. Typical test rate accuracy on the above data is around 80%; see [11] for more details on the predictive accuracy of this algorithm for TF target prediction.

Positive Examples: Known positive examples of genes g with promoters binding t include binding targets taken from ChIP-chip experiments [29], [8], Transfac 6.0 Public [24], and a list curated by Young et al., from which we have excluded indirect evidence such as sequence analysis and expression correlation [29]. Of the ChIP-chip interactions, only those with p-values 10^{-3} (i.e., high confidence level) are considered positives. Selected negatives are a randomly chosen subset of those genes found not to be bound by a TF in ChIP-chip experiments (typically these are the genes with highest p-values and thus least significant binding).

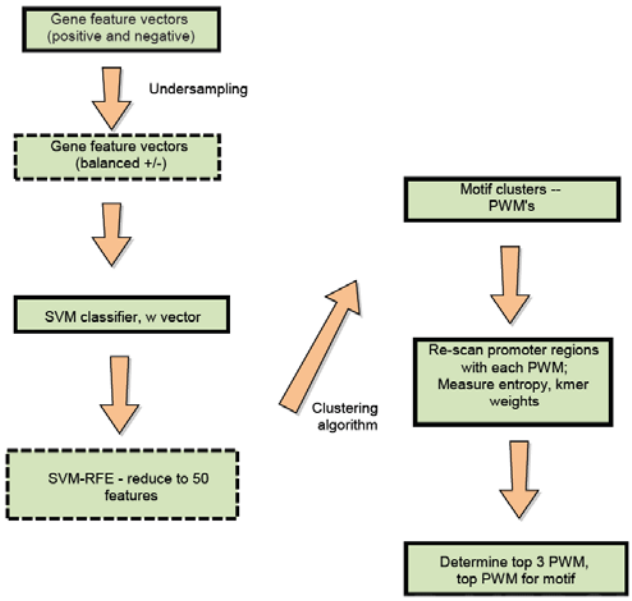

Figure 1: Workflow of the algorithm for a fixed TF t: (a) Feature vectors of genes consist of counts of 4-, 5-, and 6-mers, with certain very common k-mers excluded. Negative examples are chosen from ChIP-chip experiments [8] with high p-values. (b) Approximately 600 negative examples are undersampled to match the number of positive examples (typically 100 to 200 per promoter region). This provides balanced sets of positive and negative feature vectors with which to perform SVM discrimination. (c, d) The SVM classifier provides a weighted direction vector $\tilde{w}$, whose largest components are iteratively recalculated 20 times, from which 50 largest components (with corresponding k-mer positions in the current $\tilde{w}$ vector) are determined at each run, yielding up to 600 candidate k-mers. These are reduced using R-SVM [38] to 50. (e) These k-mers are extracted for clustering in the formation of motif PWM's. (f) The resulting candidate PWM's are evaluated by (i) re-scanning the positive promoters and determining individual matrices' scores (ii) examining entropy (purity of columns) and weights (numbers of k-mers clustered to form the PWM). (g) This information is combined into a score, from which the top PWM's are chosen.

For each gene g in *S. cerevisiae*, the total number of positive examples $n(g)$ typically is in the range of 50 to 300. Negatives can be chosen randomly (since there are a relatively small percentage of positives in the data), or from genes in ChIP-chip experiments which have large p-values. They can also be formulated as randomized versions of true positives in cases where known non-positives are not available.

R-SVM: For fixed t, once the optimized SVM classifier $f_t(\mathbf{x}) \equiv f(\mathbf{x})$ is determined, the weighted direction vector $\tilde{\mathbf{w}}$ (see above) contains the information about important k-mers. We choose the top 150 components as the first 150 of the vector $\tilde{\mathbf{w}}_\pi$, output from $\tilde{\mathbf{w}}$ using R-SVM (see above). The corresponding positions $\{\pi^{-1}(i)\}_{i=1}^{150}$ represent k-mers whose entries are the largest in $\tilde{\mathbf{w}}$, and are the features which most differentiate positives and negatives. The final choice of 50 k-mers is obtained after 20 iterations and a second R-SVM selection.

Agglomeration: Typically there are many similarities in the k-mers among the final 50, as would be expected. An agglomeration scheme is then employed to cluster those k-mers which are similar or have significant overlap. Then a PWM (position weight matrix) is formed which reflects the relative frequencies of the bases in $\mathcal{A}$ in each position. This is similar to one used in [18], where in addition higher weights are given to k-mers which correspond to earlier positions in $\tilde{\mathbf{w}}_\pi$. Starting with the first k-mer in position $\pi^{-1}(1)$ (which can be of length 4, 5, or 6) the second k-mer at $\pi^{-1}(2)$ is matched with it, and all overlaps are tested. If the match meets a certain threshold, then the overlap is kept, the k-mers are placed in the same cluster, forming the first PWM. Every time a new k-mer is added in this way, it either adds to an existing cluster or (if there are no matching clusters) forms a new one.

Comment on choices of k-mers: The performance of any clustering method depends on the quality of top ranked k-mers reported by SVM feature selection. We want to eliminate noisy k-mers before we run k-mer count.

a. Typically the reported top ranked k-mers contain a large number of irrelevant noisy ones - typical of these are, e.g., 'AAAAAN', 'ANAAAA', 'TATATA', 'ATATA', 'ACACAC'…

b. Further, if negatives must be fabricated due to lack of information about true negatives, they can be formed as permutations of positives (preserving their statistical properties). In this case SVM will often pick out regular sequences such as the above (which are not typical in the permuted sequences) and their ranking will be artificially higher.

For these reasons we have made the following changes:

1. Delete 'AAAAA+', 'ACAC+', 'TATA+'… from the original sequence.

2. Set counting resistance to 2. Thus if a sequence is …ACACAC…, the occurrence count for 'ACAC' is not 2 but 1. Similarly for …AAAAAAAA…, the occurrence count for 'AAAAA' is 1 rather than 4.

These two operations are assumed not to affect the motif components, but they do reduce such noisy k-mers.

In the case of artificial negative generation, another method for elimination of noise would involve fitting a higher order Markov background model to the whole genome and randomly generating pseudo-negative examples based on this model.

More about Clustering: A greedy method with 2 thresholds is used. Once the score of a k-mer matched to a cluster is above the in-threshold, the k-mer is added into this cluster.

The algorithm starts the PWM with a weighted count. For example, if the current k-mer is GGGTAA with weight (in the final **w** vector) of 0.230, the weighted PWM is

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & .23 & .23 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ .23 & .23 & .23 & 0 & 0 & 0 \\ 0 & 0 & 0 & .23 & 0 & 0 \end{bmatrix}$$

with the first row representing A, and the remaining rows representing C, G, and T.

After comparing all possible alignments, an incoming k-mer with the next highest weight in $\tilde{\mathbf{w}}$ (i.e., the second entry in the ordered vector $\tilde{\mathbf{w}}_\pi$) will be added into the best fitting existing cluster with an offset determined by its best fit, and the cluster's PWM is updated to incorporate the new k-mer, by addition of the weighted PWM corresponding to it

Thus if an incoming k-mer is now CGGGTC with weight 0.17, it is clear that its position must be adjusted 1 position to the left for the best match, so the PWM is updated with alignment

C	G	G	G	T	C	
	G	G	G	T	A	A

Thus we add

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ .17 & 0 & 0 & 0 & 0 & .17 \\ 0 & .17 & .17 & .17 & 0 & 0 \\ 0 & 0 & 0 & 0 & .17 & 0 \end{bmatrix}$$

transposed to the left, yielding the jogged sum

$$S_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & .23 & .23 \\ .17 & 0 & 0 & 0 & 0 & .17 & 0 \\ 0 & .4 & .4 & .4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .4 & 0 & 0 \end{bmatrix}$$

Each column's importance is stored as its column sum divided by the total weight $(.23 + .17)$:

$$[\,.43 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad .58\,]. \tag{2}$$

The final PWM is normalized. Assuming M_2 were the final adjustment to this cluster, we normalize each column:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & .58 & 1 \\ 1 & 0 & 0 & 0 & 0 & .42 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Next log odds scores with respect to the background model (in which each entry is .25) are taken with pseudocounts of .1 added to both numerator and denominator, yielding

$$\begin{bmatrix} -1.81 & -1.81 & -1.81 & -1.81 & -1.81 & .95 & 1.66 \\ 1.66 & -1.81 & -1.81 & -1.81 & -1.81 & .59 & -1.81 \\ -1.81 & 1.66 & 1.66 & 1.66 & -1.81 & -1.81 & -1.81 \\ -1.81 & -1.81 & -1.81 & -1.81 & 1.66 & -1.81 & -1.81 \end{bmatrix}$$

with each entry a log ratio (base 2). Here a 0 represents no additional information (random entry), while a positive entry represents positive information. Now multiplying by the weight vector (2) yields the final PWM,

$$W = \begin{bmatrix} -.77 & -1.81 & -1.81 & -1.81 & -1.81 & .95 & .96 \\ .70 & -1.81 & -1.81 & -1.81 & -1.81 & .59 & -1.04 \\ -.77 & 1.66 & 1.66 & 1.66 & -1.81 & -1.81 & -1.04 \\ -.77 & -1.81 & -1.81 & -1.81 & 1.66 & -1.81 & -1.04 \end{bmatrix},$$

used to compute hitting scores by scanning a promoter region with W and counting the number of scores above a given threshold T.

Intuitively, a match in an 'important' position of W should gain more of a score and a mismatch in an important position should lose more. Note that a match or mismatch at the 'ends' will count less here, since importance multiples each column.

Scoring: How do we pick out the true cluster among all clusters (each now a PWM) for t? The first part of selecting the best PWM involves the relative entropy score

$$E = \text{Total\_Weight} \cdot \sum_{j=1}^{L} r_j \sum_{i=1}^{4} p_{ij} \log \frac{p_{ij}}{b_i},$$

to pick out overrepresented clusters. Here p_{ij} is the probability (frequency) of letter i in the j^{th} PWM position, b_i is the probability of i in the background model (here .25); r_j is the importance score (see (2)) of the j^{th} column. Based on their entropy scores, we choose the top 5 clusters (and their PWM's) as candidates and use them to scan the promoter sequences which are positive for t.

The hitting ratio score for a PWM is defined as

$$\text{HR} = \frac{\text{\# hits on positive genes}}{\text{\# hits on negative genes}}$$

where negative genes are randomly undersampled to match the current number of positives. This is used to assess the quality of these 5 candidate clusters. Appearing significantly more often in positives than in negatives is assumed to be a property of a true motif. If we use fabricated negatives instead of true ones (see above) then, as suggested earlier, regular and generally overrepresented k-mers like 'AAAAGA' and 'CTTCTTCTT', can get high hitting scores in promoter regions. For this reason we have chosen to use a product of entropy and hitting ratio scores as the final criterion; the cluster with highest product is output as the best prediction.

For computation of the hitting ratio we must set a threshold T on a local PWM score to count a hit. However, it is difficult to produce common thresholds for all TFs. We have fixed the value $T = 6.4$ for all TFs, based on the heuristic fact that for a normalized PWM, a single strong match can add 1.65 ($= \log_2[(1 + .1)/(.25 + .1)]$, the maximal log odds ratio) to the PWM score, while a moderate match gains 1 and a weak match gains 0.5. Mismatches in corresponding different levels will give losses of 1.8, 1 and 0.5. A hit should have at least 4 strong matches and some matches or mismatches which on the average do not affect the hitting score.

Gapped motifs: The procedure for gapped motifs simply tries a number of allowed gap sizes, and considers the regions adjacent to the gap to be contiguous for the purpose of the SVM algorithm, similarly to what is done in BioProspector.

3. Experimental results

We chose 85 TF's from *supplemental file* 1 of binding data from MacIsaac [23], consisting of TF's with binding specificities known from various sources. These 85 are chosen from a total of 88 used as a benchmark in [23]. Three TF's (MATA1, CRZ1 and ECM22) are omitted because positive examples for them did not exist in our original database. Of these, 74 are ungapped and 11 are gapped. Of the full group of 85 TF's, 28 of these (25 ungapped and 3 gapped) also appear in the Transfac [24] database, and so are analyzed separately given their presumably more accurate resulting known binding specificities.

The 85 TF's were tested using AlignAce [30], BioProspector [21], and SVMotif. The following table compares the performances these three methods on the full MacIsaac dataset.

	AlignAce		BioProspector		SVMotif	
	Top	1 to 3	Top	1 to 3	Top	1 to 3
Ungapped (74)	19	34	27	33	29	43
Gapped (11)	-	-	3	3	6	9

Table 2: Motif finding performance on MacIsaac [23] TF's with known binding specificities among AlignAce, BioProspector and SVMotif. The 'top' PWM is the one with the highest score for each TF; the number in that column indicates how many TF's have top PWM's which coincide with the reported motif in *supplemental file* 1 of MacIsaac [23]. 1 to 3 above represents the top 3 PWM for a single TF. Each score represents the number of PWM hits of the 'top' motif as reported by MacIsaac among the union of the top 3 PWM's for all TF's in the indicated row. Each PWM is scored either 0 or 1 total depending on whether or not it coincides with reported motifs. The above indicates there were a total of 74 TF's without gapped motifs, and 11 with gapped motifs. For gapped motifs each program was given a range of 3 gap sizes to try.

Below are the results for the subgroup of the above TF's which also appear in Transfac [24].

	AlignAce		BioProspector		SVMotif	
	Top	1 to 3	Top	1 to 3	Top	1 to 3
Ungapped (25)	9	15	14	15	12	17
Gapped (3)	-	-	1	1	2	3

Table 3: Motif finding performance on MacIsaac TF's (Transfac subgroup).

Finally, we have taken the 102 *S. cerevisiae* TF's from the UCSC Genome Browser [16]. In this database there are published PWM's for 102 TF's, complied largely from [8]. These have been converted by us to consensus sequences. Again after elimination of two TF's because of a lack of positive/negative examples, the corresponding results are:

	AlignAce		BioProspector		SVMotif	
	Top	1 to 3	Top	1 to 3	Top	1 to 3
Ungapped (90)	27	37	33	39	34	49
Gapped (10)	-	-	3	3	6	8

4. Signals indicating multiple switches

We have verified, as have others [17], [8] that there are functional and statistical reasons for the multiple hits which occur in all methods that correlate motifs with TF binding. For a given TF there are cofactors which can modulate the TF itself or reinforce the activity of the TF, leading to a number of the 'additional' motifs discovered by programs such as BioProspector and SVMotif. This is added to in the multiple TF's in overlapping genomic transcription modules. Thus there are consistent appearances of 'additional' PWM's statistically associated with TF's but apparently not acting as their binding sites. We have tested some of these apparently false functional relationships (in that the TF of interest does not bind the additional sites), and verified them statistically. We have divided several TF data sets in half, and showed that 'additional' PWM's were consistently duplicated on cross-validation. This cross-functionality of motifs and the regulators which may bind them is not completely understood, though it has been examined in specific situations.

We note that prediction success rate for TF motifs increases significantly if we consider the two or three highest scoring PWM candidates for t, as opposed to just the highest scoring one. This may imply that there are typically three or less likely motifs which have a statistical (and possibly functional) significance correlated with gene expression. With this hypothesis, scoring methods by the number of hits in the top 3 is a reasonable measure of accuracy of our methods in finding validated motifs from Transfac or YeastGenome. It is shown in [8] and elsewhere that multiple binding sites in a promoter are typical in yeast, and they often can act in concert; [18] offers more verification in this direction.

5. Discussion

Kernel methods such this one as are based on assigning feature vectors to genes. Such feature space-based methods can be emulated by other feature vector classification methods such as random forests or LARS (least angle regression) [7] in ways which incorporate prior knowledge into feature spaces in an identical way. The choice of kernel in SVM is equivalent to choosing a feature map Φ into a space with Euclidean geometry. Thus any prior knowledge (e.g., phylogenetic information [36]) which can be incorporated into a kernel can also be incorporated into alternative methods such as those above, or any number of neural network-based techniques such as ART [10], by a proper adaptation of the feature map.

A remark on methodology: We remark on the effective simplicity of this process. The two basic components are identification of a statistical correlation between k-mers and gene activation (in sufficiently controlled experiments), and the inference of motif PWM's from agglomeration of k-mers which are significantly correlated with TF activation. The algorithms for both parts contribute to usefulness of the result.

This method could be further improved using ideas introduced in [32], in which k-mer selection is improved using phylogenetic information from orthologous species. As is shown there, this can be accomplished by improvement of the kernel. As suggested above, such incorporation of prior knowledge can also be accomplished, in use of random forests through proper weighting of k-mers in feature vectors using phylogenetic considerations. This will be considered in later work. Other promising feature vector-based methods include random projection methods [5], which are capable of handling large numbers of variables and determining important ones. Indeed, random projection and random variable selection methods (such as RF) can be combined with any number of other machine learning algorithms to handle feature spaces which would otherwise be prohibitively large.

Another important variation in the approach involves the choice of positives and negatives in the machine learning method. In the case of SVM we can undersample, choosing only more reliably classified upstream regions $\mathbf{x}$. A balance may need to be drawn between the choice of threshold and the lower number of examples which result.

Finally, we note that the greedy agglomeration algorithm described here can be replaced by other existing agglomeration procedures, though their effectiveness has not been studied in this application. Other procedures include that used by [25]. The step of statistical identification of k-mers associated with TF activation is what here requires machine learning methodologies.

6. References

[1] T. Bailey and C. Elkan (1994), "Unsupervised learning of multiple motifs in biopolymers using EM," *Machine Learning* **21**, pp. 51-80.

[2] P. Cliften, M. Johnston, et al., "Surveying Saccharomyces genomes to identify functional elements by comparative DNA sequence analysis," *Genome Research* **11**, pp. 1175-1186, 2001.

[3] E. Conlon, X. Liu, J. Lieb, and J. Liu, "Integrating regulatory motif discovery and genome-wide expression analysis," *PNAS* **100**, pp. 3339-3344, 2003.

[4] N. Cristianini, and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[5] Q. Ding, Dimensionality reduction and its application in machine learning, Technical Report, Boston University, 2006.

[6] S. Dwight, M. Harris, K. Dolinski, et al., "Saccharomyces Genome Database (SGD) provides secondary gene annotation using the Gene Ontology," *Nucleic Acid Research* **30**, pp. 69-72, 2002.

[7] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *Annals of Statistics,* pp. 407-499, 2003; see also http://www-stat.stanford.edu/hastie/Papers/LARS /LeastAngle_2002.ps

[8] C. Harbison, et al., "Transcriptional regulatory code of a eukaryotic genome," *Nature* **431**, pp. 99-104, 2004.

[9] F. Gao, B. Foat, et al., "Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data," *BMC Bioinformatics* **5**(1), p. 31, 2004.

[10] S. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, Reidel Press, Boston, 1982.

[11] D. Holloway, M. Kon and C. DeLisi, "Machine learning for regulatory analysis and transcription factor target prediction in yeast." *Systems and Synthetic Biology* **1**, 2007.

[12] D. Holloway, M. Kon and C. DeLisi, "In silico regulatory analysis for exploring human disease progression," preprint, 2007.

[13] P. Hong, et al., "A boosting approach for motif modeling using ChIP-chip data." *Bioinformatics* **21**, pp. 2636-2643, 2005

[14] E. Hong, R. Balakrishnan, KR Christie, MC Costanzo, SS Dwight, SR Engel, DG Fisk, et al., "Saccharomyces Genome Database"; http://www.YeastGenome.org

[15] T. Joachims, "Making large-scale SVM learning practical." *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (eds.), MIT Press, Cambridge, MA, 1999.

[16] D. Karolchik, et al., "The UCSC Genome Browser Database." *Nucl. Acids Res* **31**, pp. 51-54, 2003.

[17] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, C. Leslie, "Profile-based string kernels for remote homology detection and motif extraction," *Journal of Bioinformatics and Computational Biology* **3**, No. 3, pp. 527-550, 2005.

[18] A. Kundaje, M. Middendorf, F. Gao, C. Wiggins, and C. Leslie, "Combining sequence and time series expression data to learn transcriptional modules," *IEEE/ACM Trans Comput Biol Bioinfo.* **2**, pp. 194-202, 2005

[19] A. Kundaje, et al., "A classification-based framework for predicting and analyzing gene regulatory response." *BMC Bioinformatics* **7**.

[20] G. Lanckriet, N. Cristianini, et al., "A statistical framework for genomic data fusion," *Bioinformatics* **20**, pp. 2626-2635, 2004.

[21] X. Liu, D. Brutlag, and J. Liu, "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *Pac. Symp. Biocomput*. **6**, pp 127-138, 2001.

[22] X. Liu, D. Brutlag, and J. Liu, "An algorithm for finding protein-DNA interaction sites with applications to chromatin immunoprecipitation microarray experiments." *Nature Biotechnology* **20**, pp. 835-39, 2002.

[23] K. MacIsaac, et al., "An improved map of conserved regulatory sites for Saccharomyces cerevisiae." *BMC Bioinformatics* **7**, pp. 113-127, 2006.

[24] V. Matys, O. Kel-Margoulis, et al. "TRANSFAC(R) and its module TRANSCompel(R): transcriptional gene regulation in eukaryotes." *Nucl. Acids Res.* **34**, pp. 108-110, 2006.

[25] M. Middendorf, A. Kundaje, C. Wiggins, Y. Freund, and C. Leslie, "Predicting genetic regulatory response using classification," *Twelfth International Conference on Intelligent Systems for Molecular Biology*, *Bioinformatics* **20** Suppl 1, 2004.

[26] P. Pavlidis and W. Noble, "Gene functional classification from heterogeneous data." *RECOMB Conference Proceedings*, pp. 249-255, 2001.

[27] T. Reddy, B. Shakhnovich, and C. DeLisi, "Binding site graphs: A new graph theoretical framework for prediction of transcription factor binding sites," *PLOS Computational Biology* **3**, pp. 844-854, 2007.

[28] B. Ren, R. Young, et al., "Genome-wide location and function of DNA binding proteins," *Science* **290**, pp. 2306-2309, 2000.

[29] B. Ren, F. Robert, J. Wyrick, et al., "Genome-wide location and function of DNA-binding proteins," *Science* **290**, pp. 2306-2309, 2000.

[30] F. Roth, J. Hughes, P. Estep, and G. Church, "Finding DNA regulatory motifs within unaligned non-coding sequences clustered by whole-genome mRNA Quantitation", *Nature Biotechnology* **16**(10), pp. 939-45, 1998.

[31] B. Schölkopf, and A. Smola, *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, 2002.

[32] B. Schölkopf, K. Tsuda, and J. Vert, *Kernel Methods in Computational Biology*, MIT, Cambridge, MA, 2004.

[33] N. Simonis, S. J. Wodak, et al., "Combining pattern discovery and discriminant analysis to predict gene co-regulation," *Bioinformatics* **20**(15), pp. 2370-2379, 2004.

[34] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 2000.

[35] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1998.

[36] J-P Vert, R. Thurman and W. S. Noble, "Kernels for gene regulatory regions," *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, 2005.

[37] C. Workman and G. Stormo, "ANN-Spec: a method for discovering transcription factor binding sites with improved specificity," *Pac Symp Biocomput.,* pp. 467-78, 2000.

[38] X. Zhang, et al. "Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data," *Bioinformatics* **7**, p. 197, 2006.