Learning Methods for DNA Binding in Computational Biology

Mark Kon*, Dustin Holloway, Yue Fan, Chaitanya Sai and Charles DeLisi

Abstract — **We describe some recently developed and new applications of learning theory to prediction of binding by transcription factors to DNA in yeast and humans, as well as location of binding sites. This has potential applications to new types of biochemical bindings as well. Some related algorithms for identifying binding site locations are described.**

1. Introduction

We present here a review of recent results and some developing methods using kernel-based learning, for pairing genes with DNA regions and with specific locations involved in genetic transcription control. Identification of such pairings is a fundamental part of understanding gene expression regulation in a cell, is a key to identifying fundamental biochemical interactions and pathways, and gives basic insight into the cell's development and its response to damage or stress. At the center of this process is the direct interaction of transcription factors (TFs) and the specific locations (*cis*-elements) they bind to in DNA. The biological binding mechanism is hard to solve chemically, or predict using other non-experimental methods (fig 1 below). Note that the notion of TF binding with a gene here actually involves binding of DNA regions adjacent to the gene such as introns or the upstream or downstream region of the gene (fig. 3 in §II), though we will sometimes loosely state that a gene binds a TF. A typical HTH-type DNA binding is illustrated in fig. 1.

Methods in computational biology are making it possible to avoid experimental procedures (see, e.g., [9], [18]) for identifying gene-TF interactions, using computed implementations of new mathematical and statistical methods.

Manuscript received January 31, 2007.

Mark Kon and Yue Fan are in the Department of Mathematics and Statistics, Boston University, Boston, MA 02215 (emails mkon@bu.edu, yue@bu.edu). M. Kon is the corresponding and presenting author. Telephone: 617-353-9549.

Dustin Holloway is in the Departments of Molecular Biology, Cell Biology, and Biochemistry, Boston University, Boston, MA, USA 02215 (email dth128@bu.edu).

Chaitanya Sai is in the Department of Cognitive and Neural Sciences at Boston University, Boston, MA 02215 (email gsc@cns.bu.edu).

Charles DeLisi is in Bioinformatics and Systems Biology, Boston University, Boston, MA 02215 (email delisi@bu.edu).

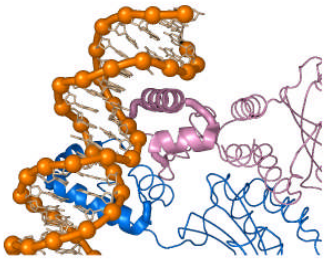


Fig. 1: DNA binding in *Agrobacterium tumefaciens* (E. Zhuang, K. Kahn, [62])

Diversity of information: Information which can be relevant to computational determination of TF-gene identifications is highly diverse, including such elements as microarray experiment correlations [27], [34], [6] gene ontology [11], and phylogeny [5] information. It is sparse, with some data available for some genes only, and very high-dimensional (DNA string counts give thousands of data dimensions; see §II).

Learning methods which extrapolate binding rules from examples have the potential for widespread and practical use. Current approaches include artificial neural network (ANN) methods (e.g., the ANN-spec algorithm [59], which combines weight-matrices and artificial neural networks; radial basis function (RBF) methods [10]; kernel learning methods, e.g., support vector machine (SVM) [58] and k-nearest neighbors (kNN) methods.

Kernel learning: In almost all cases any type of biological and computational information applied to identification of a TF binding target (i.e., a DNA location where it binds) needs to be put into a structured framework. Kernel-based learning methods, particularly support vector machines (SVMs) [56], [57], [7], [46] give a natural basis for integrating input data. The methods have been used in genomics in a number of ways, for example for predicting gene function from gene datasets [40], [28].

We complete the introduction by briefly mentioning some of the context relating kernel learning to prior neural net methods. Kernel learning methods started as applications in artificial and then RBF-based neural networks [42], [15] before their current formulations [51]. They extended into statistical and machine learning methods, and are now used by theoretical and applied workers in neural networks, statisticians, and computer scientists. Kernel learning has

expanded over 20 years as neural paradigms (e.g., Grossberg equation feedforward nets [17]) have extended to statistically oriented RBF networks, and then kernel approaches for regularization networks and SVM. Perceptrons have effectively been generalized to SVM, and feedforward neural nets now have extended to include RBF nets, with kernels based on optimization of well-defined Lagrangians. Kernel learning is now used in computer science, statistics, artificial intelligence and computational biology, as well as in neuroscience.

We remark that some of the results here are abbreviated because of space limitations. These are given in more detail in [23] and [25].

II. Biological Background, Feature Space Methods

The transformation of DNA information into working proteins in the cell involves transcription of DNA to pre-mRNA, which becomes mRNA and leads to a protein, which then determines cellular properties. We will consider only the start of this process (transcription); see fig. 2.

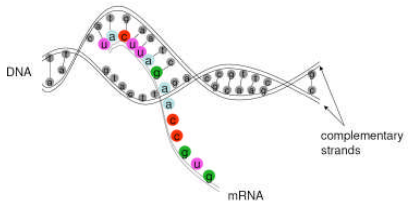


Fig. 2: Transcription: a strand of DNA is transcribed into RNA (http://biology.unm.edu/ccouncil/Biology_124/Summaries/T&T.html)

The promoter region of DNA near a given gene, is the domain to which TF's generally bind; this is the upstream region in the case of yeast (see fig. 3). It contains regulatory sequences which attract transcription factors (TF's), whose presence is required for transcription of the DNA into RNA. Regulatory sequences are inexactly repeating patterns known as motifs, which stand out as similar patterns across species - their function is to attract specific TF's. TF's bind to the promoter region at transcription factor binding sites (TFBS).

Definitions: Assume a fixed species $\mathcal{S}$ (e.g., the yeast *S. cerevisiae*) has genome $\mathcal{G}$ (collection of genes). Generally, a fixed TF t attaches to a regulatory motif, a stretch of DNA between 5 and 20 base pairs long (fig 3).

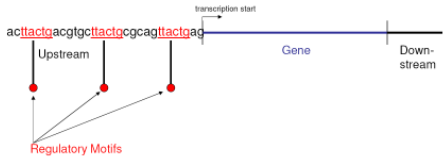


Fig. 3: Gene and upstream motif structure for yeast

To learn the rules for interaction with a given gene g (for fixed t), consider a training data set $\mathcal{D}_0 = \{(g_i, y_i)\}_{i=1}^n$, where $g_i \in \mathcal{G}$ and $y_i \in \mathbb{B} = \{-1, 1\}$. Define the correct classifier $y_i =$

$$\begin{cases} 1 & \text{if } g_i \text{ attaches TF (under any experimental condition)} \\ -1 & \text{otherwise} \end{cases}.$$

We wish to learn $f_0 : \mathcal{G} \to \mathbb{B}$ defined by

$$f_0(g) = y \ = \text{correct classification of } g$$

from the examples in $\mathcal{D}_0$. For the initial case of *S. cerevisiae*, we define the gene $g = b_1 b_2 \dots b_{800}$ formally by its upstream (binding region) sequence of 800 or less bases b_i (for a fixed orientation and side of the double helix).

Thus f_0: $\mathcal{A}^{800} \to \mathcal{B}$, with $\mathcal{A} = \{A, G, C, T\}$. For any g, form a feature vector $\phi(g)$ using information about g or its upstream region. This might be a string vector $\phi_{\text{str}}(g)$ based on a lexical enumeration $\{\text{STR}_i\}_i$ of all consecutive 6-mers (strings of length 6), e.g., $\text{STR}_{i_1} = ACAGTC$, $\text{STR}_{i_2} = CGTACA$, etc., appearing in g's upstream region. The component $\phi_{\text{str}}(g)_i$ is then the upstream count of STR_i in gene g. The feature map ϕ maps into the string feature space F_{str} consisting of possible string vectors $\phi_{\text{str}}(g)$.

Another feature map is $\phi_{\text{mot}}(g)$: $\mathcal{G} \to F_{\text{mot}}$, with $\phi_{\text{mot}}(g)$ the upstream count of occurrences of MOT_i, with $\{\text{MOT}_i\}_i$ an enumeration of 104 motifs (known binding sequences for any TF in *S. cerevisiae*). Note that motif counting is a standard way of identifying t-g binding and binding locations for t near g. Another useful map is $\phi_{\text{exp}}(g)_i$, an expression data profile for g, i.e., a Boolean array indicating expression or lack of expression of g in a set of microarray experiments.

General feature maps: Consider a general map $\phi : \mathcal{G} \to F$, with F the *feature space* (a vector space), and $\mathbf{x} = \phi(g) \in F$. If $f_0 : \mathcal{G} \to \mathbb{B} \equiv \{-1, 1\}$ is a classification map, we may wish to assign the classification -1 or 1 directly to the feature vector $\mathbf{x}$ rather than its corresponding gene $g = \phi^{-1}(\mathbf{x})$. Then, if ϕ is invertible, the composition

$$f_1(\mathbf{x}) = f_0 \circ \phi^{-1}(\mathbf{x}) : F \to \mathbb{B}$$

accomplishes this, assigning a classification to each $\mathbf{x} \in F$. In the example of the string feature space $F = F_{\text{str}}$, f_1 maps a sequence $\mathbf{x}$ of string counts in F into yes (1) or no (-1) in

$\mathbb{B}$. We replace the data $\mathcal{D}_0 = \{(g_i, y_i)\}_i$ with equivalent data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, with $\mathbf{x}_i = \phi(g_i)$. Given data (examples) $\mathcal{D}$, we seek to approximate the (unknown) exact classifier function $f_1 : F \to \mathcal{B}$ which correctly generalizes $\mathcal{D}$. For all (test) feature vectors $\mathbf{x}$, this yields $f_1(\mathbf{x}) = y$, with y the correct classification of $\mathbf{x}$. The problem of determining f_1 may be reformulated to seek $f : F \to \mathbb{R}$, where

$$f(\mathbf{x}) > 0 \text{ if } f_1(\mathbf{x}) = 1; \;\; f(\mathbf{x}) < 0 \;\; \text{if} \;\; f_1(\mathbf{x}) = -1;$$

for technical reasons it will be easier to work with procedures which (equivalently) find such an f instead.

We have used 26 feature maps (and kernels) in our yeast studies. Table 1 contains a summary of them.

#	Abbreviation	Description of Dataset
1	MOT	Motif counts in yeast
2	CON	Motif count conservation in 18 organisms
3	PHY	Phylogenetic profiles
4	EXP	Expression correlations
5	GO	GO term profiles
6	KMER	K-mer counts – 4,5,6-mers
7	S1	Split 6-mer counts 1 gap kkk_kkk
8	S2	Split 6-mer counts 2 gaps kkk__kkk
9	S3	Split 6-mer counts 3 gaps kkk___kkk
10	S4	Split 6-mer counts 4 gaps kkk____kkk
11	S5	Split 6-mer counts 5 gaps kkk_____kkk
12	S6	Split 6-mer counts 6 gaps kkk______kkk
13	S7	Split 6-mer counts 7 gaps kkk_______kkk
14	S8	Split 6-mer counts 8 gaps kkk________kkk
15	M01	6-mer counts with 1 mismatch (count = 0.1)
16	M05	6-mer counts with 1 mismatch (count = 0.5)
17	ENT	Condition specific TF-target correlation
18	BIT	Nucleotide sparse binary encoding
19	CRV	Promoter curvature prediction
20	HC	Homolog conservation
21	HYD	Hydroxyl cleavage
22	KPo	Kmer median positions from start
23	KPr	Kmer Probabilities (-log *p*-val)
24	MT	Promoter Melting Temperature-20bp window
25	DG	Promoter Melting ΔG profile-20bp win
26	BND	Promoter bend prediction
Random Control	R	Scrambled KMER data
Random Control	RN	Random normally distributed data
Random Control	RH	Scrambled Random selection of 10% of each dataset

Table 1: List of feature maps used for *s. Cerevisiae* binding prediction. k-mer denotes a string of length k [23]. MT and DG are calculated using the EMBOSS1 toolbox. EMBOSS uses the nearest-neighbor thermodynamics from [44], [12]. KMER: Background k-mer (string of length k) counts in the upstream region are calculated with RSA tools [53], [54]. Our method for calculating related probabilities is similar to that described in [55]. HYD: a database of DNA sequences and their hydroxyl cleavage patterns has been published [38]. This database allows accurate prediction of DNA backbone accessibility for any sequence by sequentially examining every 3-mer in a sequence and looking up its experimental cleavage intensity as measured by phosphor imaging of cleaved, radio-labelled DNA separated by electrophoresis [1]. BND, CRV: Using the Banana algorithm in the EMBOSS [44], [12] toolkit, bend and curvature predictions were made for all yeast promoters. Banana follows the method of [16] which is consistent with experimental data [48].

We use SVM on training data $\mathcal{D}$ to predict binding on test data (fig. 5). Precisely, we encode the feature data $\phi(g_i) = \mathbf{x}_i$ into a kernel function $K(\mathbf{x}, \mathbf{y})$, representing an imposed geometry on F in which $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ is the inner product of its arguments (see §III).

Many of these data sets (each of which yields a different feature map F_k and kernel K_k) yield weak classifiers at best, but they can combine using kernel methods to give better predictions (§III). The figure below depicts predictive accuracy for each kernel individually, and then the combination of all kernels. Performances in terms of sensitivity (dashed), positive predictive value (PPV, solid) and F1 score (blue, the harmonic mean of sensitivity and PPV) are given. The binding data are from [61], [20], [30], [33].

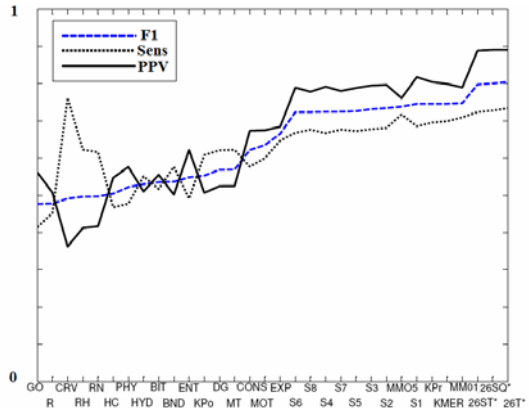


Fig. 4: Predictive accuracies of the 26 kernels above using SVM to predict binding of genes g with a fixed TF t (averaged over t). PPV is positive predictive value, while $F1$ is the harmonic mean of PPV and sensitivity (see [23]).

III. SVM Learning and the Kernel Trick

Here is a brief overview of the SVM learning approach. The SVM provides a discriminant function f of the form $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, with $f(\mathbf{x}) > 0$ yielding the classification $y = 1$ (indicating $g = \phi^{-1}(\mathbf{x})$ binds t) and otherwise $y = -1$. The optimal $\mathbf{w}$ can be found using quadratic programming, from which it is shown $\mathbf{w} = \sum_i \alpha_i \mathbf{x}_i$ is a linear combination of data. Thus

$$f(\mathbf{x}) = \sum_i \alpha_i \mathbf{x}_i \cdot \mathbf{x} + b.$$

What about nonlinear separators f? In F we have a set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of training vectors $\mathbf{x}_i$ for which binding y_i is known. As mentioned above, we can define a new geometry on F by replacing the standard dot product $\mathbf{x} \cdot \mathbf{y}$ with $K(\mathbf{x}, \mathbf{y})$ for an appropriate kernel K. This changes the separator

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

into a nonlinear one, allowing very general nonlinear separating surfaces on F. Again the α_i are found using linear algebra involving the kernel matrix $K_{ij} \equiv K(\mathbf{x}_i, \mathbf{x}_j)$.

An example of K is the Gaussian kernel, for which

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b = \sum_i \alpha_i e^{-\frac{|\mathbf{x}-\mathbf{x}_i|^2}{2\sigma^2}} + b.$$

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 kernel trick: Kernel learning algorithms are equivalent to RBF networks with particular choices of RBF functions. There are three distinct reasons why these methods are useful in biological and other applications. First, kernels allow us to modify the geometry of our biological feature space. Second, they allow us to work in the minimal dimension necessary to deal with our data - a dimension far lower than that of the biological feature space. Indeed, the kernel matrix K_{ij} has the dimension of the data cardinality, which may be far less than the dimension of $\mathbf{x} \in F$. Finally kernels incorporate a priori biological information (e.g., disparate information sources mentioned in §I) and combine it in the easiest possible way - concatenation of biological feature spaces F_a and F_b is equivalent to summing the corresponding kernels, yielding

$$K_{ab} = K_a + K_b.$$

In our case, the concatenation of the above 26 feature spaces (datasets) is accomplished by taking a linear combination of the 26 associated kernels (§II above).

IV. Probabilistic Interpretation of RBF Outcomes and Applications

Probabilistic approaches to SVM include a posterior probabilistic SVM (PSVM), which gives p-values (confidences) for classifying genes using the correlated parameter $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ (the SVM score). Specifically [41], the probability score $s(\mathbf{x}) = P(y = 1|f(\mathbf{x}))$ has an empirical distribution which can be used, though only if f (i.e. $\mathbf{w}$ and b) is determined by a training set fully independent of the separate training set determining $s(\mathbf{x})$. The result is an empirically based confidence level for SVM predictions. This can be used to generalize from known examples (by a factor of 10, for high-confidence genes) the known binders to some human transcription factors, with interesting implications, e.g., for biochemical pathways such as carcinogenesis pathways (§V).

Running the algorithm: The overall structure of the algorithm as it is run on yeast is:

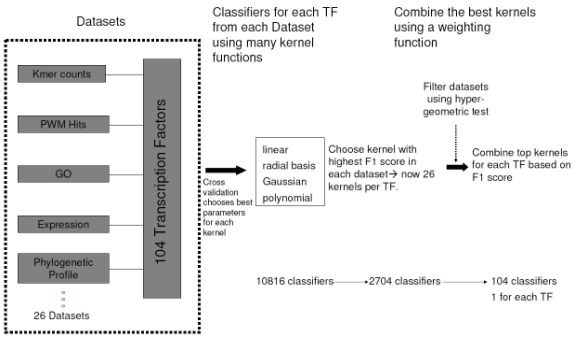


Fig. 5: The SVM algorithm uses all 26 kernels in a weighted combination with cross-validation determining parameters. Care is taken to assure there is no overfitting due to an excess of free weight parameters. The kernel coefficients β_k (below) are based on the F_1 score (harmonic mean of the sensitivity and PPV) of the kernel; see [23].

We have studied a total of 163 TF's in yeast, using combination kernels involving weighted sums of the 26 feature kernels (Table 1), and a final computational kernel (fig. 5)

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{26} \beta_i K_i(\mathbf{x}, \mathbf{y}).$$

with β_i determined by the F_1 score of kernel K_i on its own. More specifically, β_i can be the scaled F_1 score, square of the scaled F_1 score, or the squared tangent of the F_1 score (see results in last 3 data points in Fig. 4). The latter weightings emphasize higher and better F_1 values more.

To summarize the predictive values of the SVM classifiers, the best single kernel has an overall (averaged over 163 TF's) sensitivity of .71 and PPV of .82. The squared-tan weighting gives a sensitivity of .73 and PPV of .89.

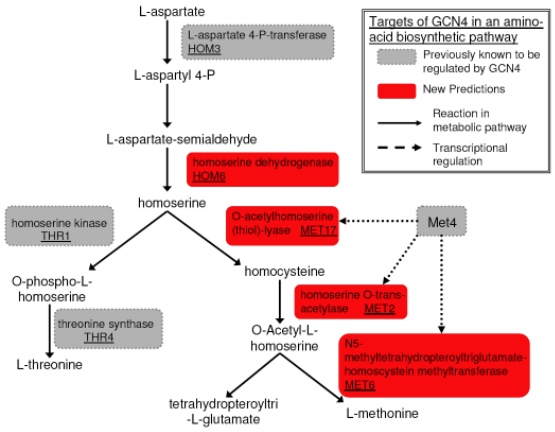


Fig. 6: Pathway predictions for the yeast TF GCN4 based on SVM predictions [23]

Fig. 6 includes new implications formed by the yeast binding predictions applied to GCN4 biochemical pathways related to amino acid biosynthesis.

V. Predictions for the Human Genome

Preliminary work using PSVM [41] modeling has been applied to make target predictions for 163 human TFs. The results are promising, with the top 33 TFs reaching a combined predictive precision of better than 60%. All predictions are available online ([22]; http://cagt10.bu.edu/TFSVM/Main%20Frame%20Page.htm). We note that the applications of SVM to human genomes described here use linear SVM kernels only. More details will appear in [25].

Wt1 gene predictions and Wilms' tumor: WT1 is a TF involved in Wilms' Tumor, making up 8% of childhood cancers [29]. This can develop in numerous ways, including loss of the WT1 producing gene (denoted *WT1*), loss of other chromosomal loci, and gene duplication. SVM predictions for WT1 allow us to suggest new Wilms tumor models. Genes in significant loci include several oncogenes and tumor suppressors which are candidates for involvement in cancer progression and may partially explain the observed clinical and biochemical data on this cancer. One example of this can be seen in chromosomal region 11p15.5, which is known to be involved in Wilms' Tumor. Newly predicted targets for WT1 are statistically enriched ($p = 6.3e$-5) for genes falling in this region and three are possible tumor suppressors, i.e., *RNH1* [14], *IGF2AS* [60], and *CD151* [49]. Other regions known to play a role in Wilms' Tumor also contain new target predictions (16q, 1p36.3, 16p13.3, 17q25, and 4p16.3). The anti-apoptotic effects of WT1 are also reviewed in [25] along with several new target genes, including *BAX* and *PDE4B*, which may help mediate the above effect. Finally, motif discovery is used to propose a new binding motif for WT1 which will be useful in later site identification. Fig. 7C below shows the top three motifs reported by the Weeder motif discovery algorithm.

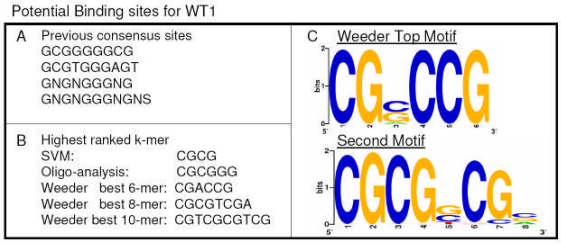


Fig. 7 - Wt1 target motifs: (**A**) Suggested consensus binding sites from the literature: references for known binding sites of WT1: GCGGGGGCG [43], GNGNGGGNG [13], GNGNGGGNGNS [21], and GCGTGGGAGT [35]. (**B**) Rankings of candidate motif strings as determined by application of SVM to a string feature space F_{str} (see also §VII; see [54] for Oligo-analysis). RSAtools: [53] (**C**) Top ranked motifs using the Weeder algorithm [25]. Logos obtained using [50].

VI. Synopsis Kernels: Learning on Learning

As mentioned in §III, kernel summation $K = \sum_{i=1}^{26} \beta_k K_k$ corresponds to a direct sum (concatenation) of feature spaces $\bigoplus_{k=1}^{26} F_k$ and so represents taking a union of all their coordinates. Such coordinate concatenations may or may not be effective. If data are sparse compared to their dimensionality (as for string kernels), kernel addition and coordinate concatenation may not be advisable, given dimensionality will be further increased. For example, in the case of addition of string and gapped string kernels (gapped string feature maps count strings but ignore fixed size gaps in them), the dimensionality of the feature space doubles from approximately 5,000 to 10,000, giving a highly sparse space. Any set of training data with size lower than this dimension can be separated by some hyperplane, and overfitting is a serious risk. It is better to initially reduce dimension of each component kernel K_k to a smaller relevant set of coordinates, with cardinality smaller than the data if possible.

The so-called synopsis kernel can be used do this. Use of this kernel gives much greater control of the coordinate optimization process than does simple kernel addition and coordinate concatenation. For each feature space F_k, only one dimension (the direction of the SVM classification gradient $\mathbf{w}_i$) is used in constructing the synopsis feature space, yielding a final feature space whose dimension equals no more than the cardinality of different feature spaces F_k originally used. Thus the only coordinates the synopsis SVM uses are projections onto the n weight vectors $\{\mathbf{w}_k\}$,

one for each feature space F_k. This reduces the analysis to arguably the most sensitive n-dimensional subspace (span of $\{\mathbf{w}_k\}$) of the full feature space $F = \bigoplus_{k=1}^{n} F_k$.

The data below are given for the yeast TF GCN4, using 8 weak classifiers, none of which have an intrinsic accuracy greater than 55%. These include phylogenetic profiling, hydroxil cleaveage, and promoter bend prediction (fig. 8).

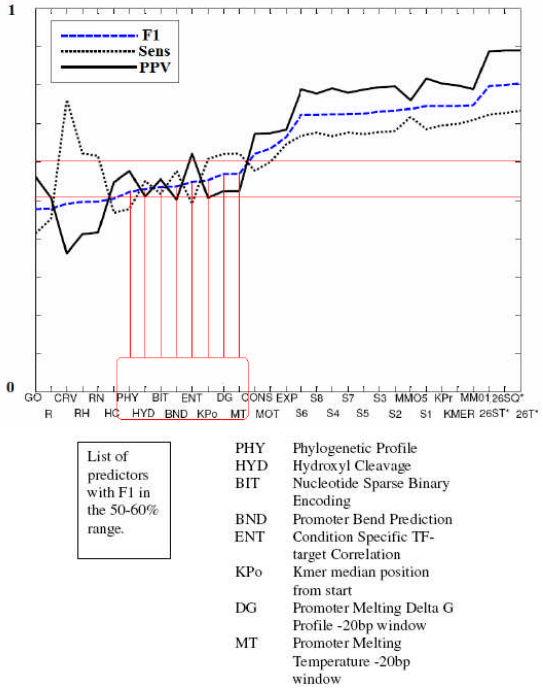


Fig. 8: 8 weak classifiers are chosen in a test of combining their synopsis vectors.

The SVM is run on each of the eight kernels, producing a gradient vector $\mathbf{w}_k$ for each corresponding feature space F_i. We then restrict our features to the 8 dimensional subspace of $\bigoplus_{k=1}^{8} F_k$ spanned by $\{\mathbf{w}_k\}_{k=1}^{8}$, and run the SVM on this. In 8 dimensions of course we can also form new nonlinear coordinate combinations (there are 36 quadratic combinations of these 8 coordinates) without risk of overfitting.

Though the predictive error rates of each weak classifier are greater than 45%, a strong synergy occurs among them when their synopsis vectors are combined on the GCN4 data set (this set has a total 211 positive gene examples, which are split into training and test data). Using just the synopsis vectors, we have the error rates:

	linear kernel	RBF kernel	quad. kernel
Error rate	.3784	.4324	.3986

From this we see dimensional reduction via synopsis vectors can be a valuable error reduction tool, given the unreduced error rate above of 45%. Here dimension has been pruned to one per kernel, and naturally more single dimensions can be added from the 26 feature spaces, based on the right criteria. What are the criteria? One method which shows promise combines optimizing discriminatory ability with stochastic independence for the coordinates. This can be done for each feature map by choosing a first coordinate to be the objective function $f(\mathbf{x})$ of the optimal SVM, and the next several coordinates based on a Gram-Schmidt orthogonalization process, with inner product determined by the empirical covariance matrix of the data themselves.

VII. Motif Finding

The above SVM learning methods can also be used for identification of binding sites using the following approach. This is illustrated briefly for humans in the identification of WT1 binding sites in §V (Fig. 7). This strategy can be organized into a motif-identification algorithm based on identification of appropriate position weight matrices (PWM).

The idea is based on the fact that the SVM separates two groups of data (positive and negative) in feature space. The present algorithm uses the string space F_{str}, by identifying the largest components (each of which corresponds to a unique string) of the gradient vector $\mathbf{w}$ which differentiates binders from non-binders in F_{str}. These largest components are identified and clustered into groups of overlapping sequences. For each cluster, a representative PWM is be derived

The derivation of a probability weight matrix (PWM) from a set of overlapping motifs (here for the TF GCN4) is illustrated in fig 9 (here using strings obtained from using other algorithms):

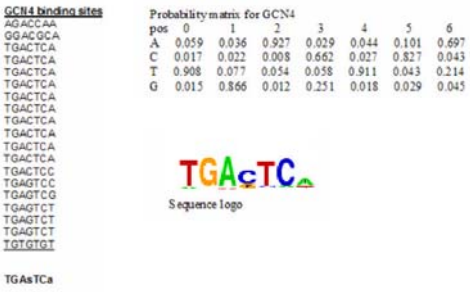


Fig 9: Known binding sites for the TF GCN4 in yeast, together with a probability matrix and a consensus sequence logo [52], [18].

Thus for our SVM-based motif algorithm, we map gene g into $\phi_{\text{str}}(g) \in F_{\text{str}}$ (its upstream string count vector) and use SVM in this case with a linear kernel function on F_{str} to separate

$\phi(g_i)$ from $\phi(g_j)$ if g_i is a binder and g_j is a non-binder. For GCN4 there are 211 positive examples and 177 (heuristically selected) non-binders.

The pseudo-code is:

Let S be the set of top strings in **w**. Let C be the current set of clusters (groups of similar strings in S).

Initial step: $S = \{\text{str1, str2, ...}\}$, ordered by significance w_i. C is initially empty.

Step 0: Form a new string cluster consisting of str1. Remove str1 from S.

Step 1: If $S =$ empty, then quit. Otherwise, pick out the string s currently in S with highest weight.

Step 2: Compute overlap scores of the string s from Step 1 with each of current clusters (represented by PWM) in C. If the highest score is greater than the adding threshold $T1$, go to step 3 (addition). If the highest score is greater than the new cluster threshold $T2$, go to step 4. Otherwise, go to step 5.

Step 3 (addition): Add string s into the cluster producing the highest score, and delete s from S. Let $S = S \cup E$ (defined in step 5). Go to step 3'.

Step 3' (deletion): Examine each element in the cluster being updated in step 3 by computing its score with respect to the empirical PWM of this cluster. If the score is smaller than the deleting threshold, move this string back into S.

Step 4: Form a new cluster in C consisting of string s, and delete string s from S. Let $S = S \cup E$. Go to step 1.

Step 5: Move string s into the exception set E. Go to step 1.

In a comparison of the SVM method to AlignACE [45] out of 30 TF's chosen with more than 150 known binding genes each, AlignACE discovered consensus motifs in 43% of instances, while the SVM method was successful in 56% of them. In the table below are comparisons of several known consensus sequences in *S. cerevisiae*, and the resulting motifs using our kernel method, versus several other standard methods (MDscan [32] and AlignACE [45] and MEME [2]). The consensus sequences below are from [20].

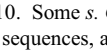
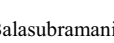
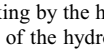
TF Name	Known consensus sequence	AlignACE	MDscan	MEME	SVM
REB1		NCGGGTAAYR	CGGGTAAT	CCGGGTAAAG	CGGGTAA
RPN4		WTTTGCCACC	ATTATT	TTTGCCACCG	TTTGCCACC
MBP1		RACGCGWA	RACGCGTC	RACGCGW	CGAAACGCGT
YDR026C		TTACCCGGMM	TTTACCCGGC	TTTACCCGGM	ATTTACCCGG
GCN4		TGAGTCAT	TGAGTCATCG	TGAGTCA	CTGAGTCATC

Fig. 10. Some *s. Cerevisiae* transcription factors, their consensus sequences, and their predictions by several algorithms.

REFERENCES

[1] B. Balasubramanian, W. Pogozelski and T. Tullius, "DNA strand breaking by the hydroxyl radical is governed by the accessible surface areas of the hydrogen atoms of the DNA backbone." *Proceedings of the National Academy of Sciences* **95** pp. 9738-9743, 1998.

[2] T. Bailey and C. Elkan (1994), "Unsupervised Learning of Multiple Motifs in Biopolymers using EM," *Machine Learning* **21**, pp. 51-80.

[3] F. Baldino, F., "High-resolution in situ hybridization histochemistry," *Methods in Enzymology* **168**, pp. 761-777, 1989.

[4] K. Breslauer, R. Frank, H. Blocker and L. Marky, "Predicting DNA Duplex Stability from the Base Sequence," *PNAS* **83**, pp. 3746-3750, 1986.

[5] 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.

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

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

[8] G. Crooks, G. Hon, J. Chandonia, S. Brenner, "WebLogo: A sequence logo generator", *Genome Research* **14**, pp. 1188-1190, 2004.

[9] B. Deplancke, D. Dupuy, et al., "A gateway-compatible yeast one-hybrid system," *Genome research* **14**, pp. 2093-2102, 2004.

[10] 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**, pp. 204, 2005.

[11] 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.

[12] Emboss web site: http://emboss.sourceforge.net/apps/banana.html

[13] G. Fraizer, Y. Wu, S. Hewitt, T. Maity, C. Ton, V. Huff, and G. Saunders, "Transcriptional regulation of the human Wilms' tumor gene (WT1). Cell type-specific enhancer and promiscuous promoter," *J. Biol. Chem.* **269**, pp. 8892-8900, 1994.

[14] P. Fu, J. Chen, Y. Tian, T. Watkins, X. Cui, et al., "Anti-tumor effect of hematopoietic cells carrying the gene of ribonuclease inhibitor." *Cancer Gene Therapy* **12**, pp. 268-275, 2005.

[15] F. Girosi, "Regularization theory, radial basis functions and networks," in *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, V. Cherkassky, J.H.Friedman, and H. Wechsler, eds., Springer-Verlag, 1994.

[16] D. Goodsell and R. Dickerson, "Bending and curvature calculations in B-DNA," *Nucl. Acids Res.* **22**, pp. 5497-5503, 1994.

[17] S. Grossberg, *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*, Reidel Press, Boston, 1982.

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

[19] C. Harbison, E. Fraenkel, R. Young, et al, http://jura.wi.mit.edu/fraenkel/download/release_v24/final_set/Final_InTableS2_v24.motifs, 2006.

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

[21] S. Hewitt, G. Fraizer, et al., "Differential Function of Wilms' Tumor Gene WT1 Splice Isoforms in Transcriptional Regulation," *J. Biol. Chem.* **271**, pp. 8588-8592, 1996.

[22] D. Holloway, "Transcription factor binding site detection by machine learning." http://cagt10.bu.edu/TFSVM/Main%20Frame%20Page.htm, 2007.

[23] D. Holloway, M. Kon and C. DeLisi, "Machine learning for regulatory analysis and transcription factor target prediction in yeast," *Systems and Synthetic Biology* **1**, pp. 25-46, 2006.

[24] D. Holloway, M. Kon and C. DeLisi, "Classifying Transcription Factor Targets and Discovering Relevant Biological Features", preprint, 2007.

[25] D. Holloway, M. Kon and C. DeLisi, "In Silico Regulatory Analysis for Exploring Human Disease Progression," preprint, 2007.

[26] 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.

[27] 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

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

[29] B. Lee, and D. Haber, "Wilms' tumor and the WT1 gene," *Experimental Cell Research* **264**, pp. 74-79, 2001.

[30] T. Lee, N. Rinaldi, et al., "Transcriptional Regulatory Networks in Saccharomyces cerevisiae," *Science* **298**, pp. 799-804, 2002.

[31] C. Leslie, E. Eskin and W. Noble, "The spectrum kernel: a string kernel for SVM protein classification," *Proceedings of the Pacific Symposium on Biocomputing*, 2002.

[32] 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.

[33] 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.

[34] 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.

[35] H. Nakagama, G. Heinrich, J. Pelletier and D. Housman, "Sequence and structural requirements for high-affinity DNA binding by the WT1 gene product. Mol. Cell. Biol.," **15,** pp. 1489-1498, 1995.

[36] P. Niyogi and F. Girosi, "On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions," *Neural Computation* **8**, pp. 819-842, 1996.

[37] W. Noble, "Support vector machine applications to computational biology," in *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, 2003.

[38] S. Parker, J. Greenbaum, G. Benson and T. Tullius, "Structure-based DNA sequence alignment," Poster: 5th International Workshop in Bioinformatics and Systems Biology, 2005.

[39] G. Pavesi, P. Mereghetti, G. Mauri and G. Pesole, "Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes," *Nucl. Acids Res.* **32**, pp. 199-203, 2004.

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

[41] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers* (A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds.), MIT Press, pp. 61-74, 2000.

[42] T. Poggio, and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science* **247** pp. 978-982, 1990.

[43] F. Rauscher, J. Morris, O. Tournay, D. Cook, and T. Curran, "Binding of the Wilms' tumor locus zinc finger protein to the EGR-1 consensus sequence," *Science* **250**, pp. 1259-1262, 1990.

[44] P. Rice, I. Longden and A. Bleasby, "EMBOSS: The European molecular biology open software suite," *Trends in Genetics* **16**, pp. 276-277, 2000.

[45] F. Roth, J. Hughes, P. Estep, and G. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation," *Nat. Biotechnol.* **16**, pp. 939-945.

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

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

[48] S. Satchwell, H. Drew, and A. Travers, "Sequence periodicities in chicken nucleosome core DNA," *J Mol Biol* **191**, pp. 659-675, 1986.

[49] G. Saur , C. Kurzeder, R. Grundmann, R. Kreienberg. R. Zeillinger et al., "Expression of tetraspanin adaptor proteins below defined threshold values is associated with in vitro invasiveness of mammary carcinoma cells." *Oncology Reports* **10**, 2003.

[50] T. Schneider and R. Stephens, "Sequence Logos: A New Way to Display Consensus Sequences," *Nucleic Acids Res.* **18**, pp. 6097-6100 1990.

[51] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.

[52] G. Stormo, "DNA binding sites: representation and discovery," *Bioinformatics* **16**, pp. 16-23, 2000.

[53] J. van Helden, "Regulatory sequence analysis tools," *Nucleic Acids Research* **31**, pp. 3593-3596, 2003.

[54] J. van Helden and J. Collado-Vides, "Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies," *Journal of Molecular Biology* **281**, pp. 827-842, 1998.

[55] J. van Helden, "Metrics for comparing regulatory sequences on the basis of pattern counts," *Bioinformatics* **20**, pp. 399-406, 2004.

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

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

[58] 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.

[59] 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.

[60] J. Yang, W. Chen, Z. Liu, Y. Luo, and W. Liu, "Effects of insulin-like growth factors-IR and -IIR antisense gene transfection on the biological behaviors of SMMC-7721 human hepatoma cells," *J. Gastroenterology and Hepatology* **18**, 2003.

[61] "Young Lab Web Data: http://staffa.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=evidence."

[62] E. Zhuang, K. Kahn, UCSB, http://www.chem.ucsb.edu/~molvisual/