

# Learning probabilistic models of *cis*-regulatory modules that represent logical and spatial aspects

Keith Noto\* and Mark Craven

Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706, USA

## ABSTRACT

**Motivation:** The process of transcription is controlled by systems of factors which bind in specific arrangements, called *cis*-regulatory modules (CRMs), in promoter regions. We present a discriminative learning algorithm which simultaneously learns the DNA binding site motifs as well as the logical structure and spatial aspects of CRMs.

**Results:** Our results on yeast datasets show better predictive accuracy than a current state-of-the-art approach on the same datasets. Our results on yeast, fly and human datasets show that the inclusion of logical and spatial aspects improves the predictive accuracy of our learned models.

**Availability:** Source code is available at <http://www.cs.wisc.edu/~noto/crm>

**Contact:** [noto@cs.wisc.edu](mailto:noto@cs.wisc.edu)

## 1 INTRODUCTION

Eukaryotic gene transcription is controlled by multiple factors, which to bind to DNA in a specific arrangement in a gene's promoter region. This type of regulation system is called a *cis*-regulatory module (CRM). Such a module consists of (1) specific sequences of DNA called motifs to which transcription factors bind, (2) logical relationships between these sites and (3) spatial relationships between these sites. Three examples of logical relationships are as follows:

- AND logic; multiple required binding sites,
- OR logic; a set of motifs, any of which satisfies a binding site,
- NOT logic; a binding site which must not appear in a promoter sequence.

Three examples of spatial relationships are as follows:

- Order preference; e.g. binding site *A* appears upstream of *B*,
- Distance preference; e.g. binding site *A* appears ~125 bp from *B*, or binding site *A* appears somewhere within 50 bp from the estimated start of transcription,
- Strand preference; e.g. binding site *A* appears on the template DNA strand (as opposed to the transcribed strand).

Given (1) a set of positive DNA sequence examples thought to contain a hidden CRM, and (2) a set of negative DNA sequence examples thought not to contain the CRM, the task we consider is to learn a representation of a CRM which distinguishes between the positive and negative sequences.

\*To whom correspondence should be addressed.

Several methods have been proposed for this task (Aerts *et al.*, 2003; Beer and Tavazoie, 2000; Keles *et al.*, 2004; Segal and Sharan, 2004; Sinha *et al.*, 2003; Zhou and Wong, 2004). These methods either learn motifs, or learn rich representations of the relationships between candidate motifs, but not both. For instance, the approach of Keleş *et al.* (2004) employs a rich representation of the logical relationships between motifs. The approach of Beer and Tavazoie (2004) involves logical aspects and spatial constraints regarding order, distance and strand. However, these approaches finalize motifs during a pre-processing step, before logical and spatial aspects are considered. The approaches of Segal and Sharan (2004) and Zhou and Wong (2004) learn motifs, and they do represent that binding sites must appear relatively close together, but in a window of a fixed and predetermined size.

Since motifs are hidden in data, learning the relevant logical and spatial aspects may help to identify motifs which would not otherwise appear to be significantly overrepresented. None of the aforementioned approaches learn both binding site motifs as they learn the logical and spatial aspects of a CRM. The approach we present is, to our knowledge, the first learning algorithm which does so.

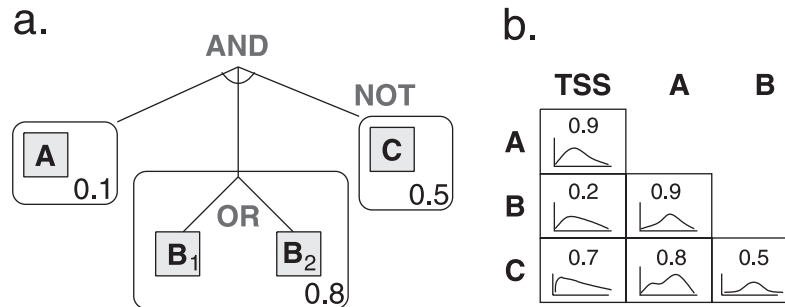
Another advantage of our approach is that it can take advantage of background knowledge. Learning CRMs is a difficult problem in part because training set sizes are often limited (for instance, they may consist of a few genes which are upregulated together) and because the relevant binding site motifs may appear anywhere in a large control region of a gene (tens of thousands of base pairs in higher eukaryotes). Since our approach is entirely probabilistic, a prior probability distribution over where binding sites are likely to be [e.g. from a set of multiple alignment conservation scores or from data concerning hypersensitive regions (Noble *et al.*, 2005)] fits naturally into our approach.

## 2 APPROACH

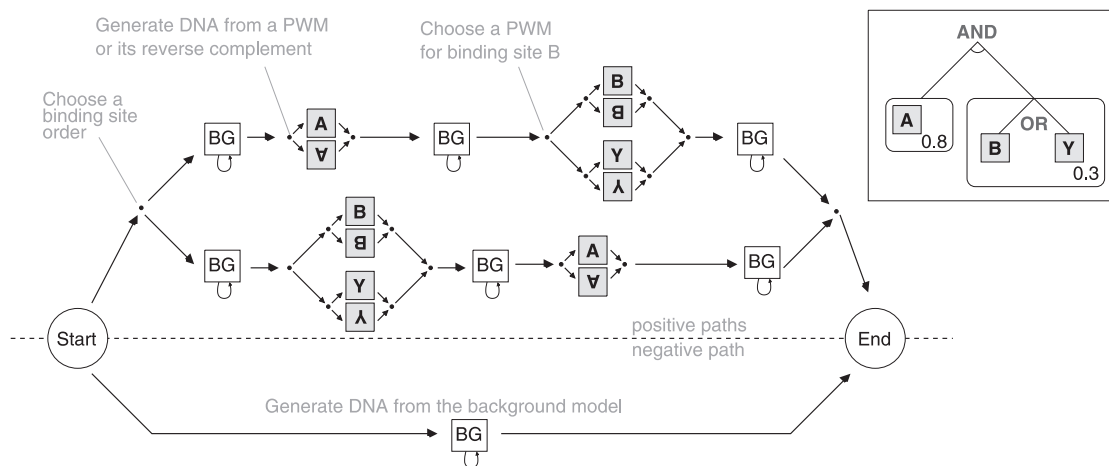
### 2.1 CRM representation

Our representation of a CRM is illustrated in Figure 1. It can describe a logical relationship among binding site motifs (using AND, OR and NOT), each represented by the standard position weight matrix (PWM), and a set of probabilities representing the spatial aspects: order, distance and strand. Note that we make a distinction between binding sites and motifs, since because of OR logic, more than one motif may be associated with a given binding site. Also note that the pairwise distance distributions in Figure 1b refer to adjacent binding sites. Hence, the order of binding sites determines which distance distributions are relevant.

It is useful to think of an instantiation of our CRM representation as a hidden Markov model (HMM), such as the one shown in



**Fig. 1.** An example model in our CRM representation. (a) A logical structure of motifs consisting of a conjunction of binding sites (rounded boxes), each with a disjunction of motif PWMs (lettered boxes). Each binding site is associated with a strand preference (one number indicating the probability of binding to the template DNA strand). A binding site may be negated. (b) Each pair of binding sites is associated with an order preference (one number indicating the probability of one site being upstream of the other) and a distance distribution over the possible intermotif distances of two adjacent binding sites. Also, each binding site has order and distance preferences relative to a fixed point, such as the estimated transcription start site (TSS).



**Fig. 2.** A DNA-generating hidden Markov model representing the logical structure shown in Figure 1 (inset). BG represents a fixed background submodel. Shaded lettered boxes represent PWMs (these are upside-down for reverse complement distributions). A sequence with the CRM takes the upper path, and generates DNA from PWMs and the background distribution. A sequence without a CRM takes the lower path, and generates all DNA from the background distribution. Note that several parameters are tied together. All instances of the background submodel are identical, and the submodel corresponding to each binding site appears multiple times for each possible binding site order.

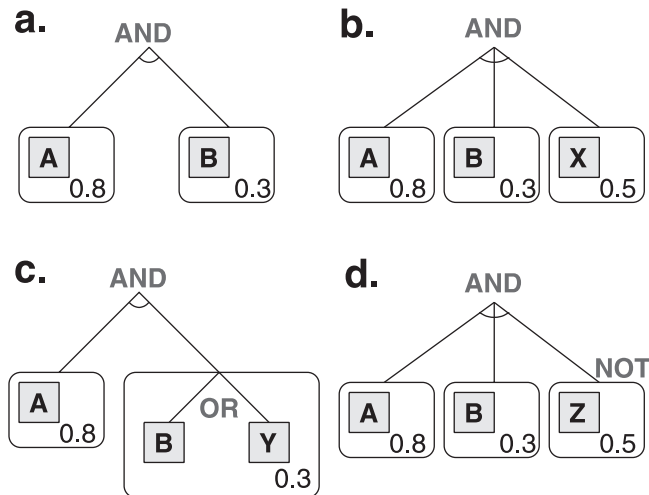
Figure 2, and each sequence as being generated by a path through this model. The logical structure of a CRM representation is reflected in the structure of the HMM. The submodel denoted ‘BG’ represents a fixed background distribution over DNA sequences of arbitrary length. We use a 5th-order HMM as the background submodel, which is trained on the appropriate DNA, such as promoter sequences in the organism being analyzed. A DNA sequence with a hidden CRM takes an ‘upper’ path through the model, and does the following: (1) probabilistically chooses an ordering of the binding sites, (2) generates DNA from a background distribution, at some point (3) chooses a PWM from which to generate the first binding site, (4) chooses a DNA strand, (5) generates DNA probabilistically from the PWM distribution, (6) generates more DNA from the background distribution, etc. The amount of sequence generated by the background distribution submodel between one binding site and the next depends on the probability of that distance, which is given by our model parameters (Figure 1b). This means that the HMM is a generalized HMM

(Burge and Karlin, 1997), because the amount of sequence explained by the background state is not represented by transition probabilities alone. A DNA sequence without a hidden CRM takes the ‘lower’ path in the model, and it generates all DNA from the background distribution.

Note that several parameters are tied together: all instances of the background submodel are identical, and the submodel corresponding to each binding site appears multiple times for each possible binding site order. Also note that the model illustrated in Figure 2 contains no negated binding sites (NOT logic). These cases are slightly different and are discussed in Section 2.5.

## 2.2 Learning structure

The task of learning one of our CRM models involves learning both the logical structure (such as the example shown in Figure 1a) and the parameters for a given structure (which is discussed in the next section). Given a structure and parameters, we evaluate a model by how well it classifies the training data (when there is a large number



**Fig. 3.** Illustration of our search space operators. (a) An initial CRM logical structure. (b) The result of applying the AND operator to the initial structure. This introduces a new binding site with an untrained PWM  $X$ . (c) The result of the OR operator. The second binding site becomes a disjunction of the original PWM  $B$  and an untrained PWM  $Y$ . (d) The result of the NOT operator. This is the same as the AND operator, except the new binding site is negated.

of training examples, we prefer to use held-aside tuning data for evaluation), according to a user-defined metric.

We search for the CRM logical structure using a best-first beam search (Mitchell, 1997). Our search operators are shown in Figure 3. We start with a structure containing a single motif. At each step in the search process, we apply each operator to the highest-scoring solution to obtain new logical structures. We learn the parameters for these structures, evaluate the models and repeat until some maximum number of structures has been evaluated.

Each application of our search operators adds a new, untrained, PWM. The model parameters we learn depend on the values in this PWM, so we initialize it by sampling from the training data.

### 2.3 Learning parameters

Given the logical structure of a candidate CRM model, we set the parameters,  $\Theta$ , in an attempt to optimize

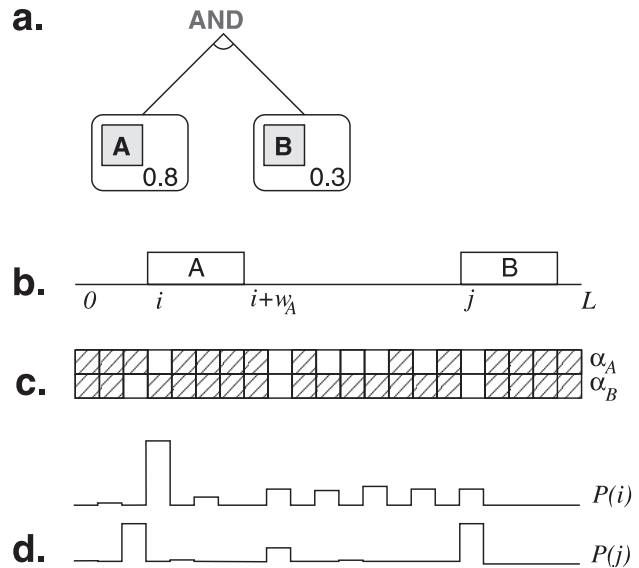
$$\hat{\Theta} = \arg \max_{\Theta} \prod_x P(c_x | x, \Theta, V_x), \quad (1)$$

where  $c_x$  is the label (i.e. positive or negative), of a training example sequence  $x$ , and  $V_x$  is a prior probability distribution describing locations on  $x$  where binding sites are likely to occur.

To train our parameters, we use the discriminative learning approach of Krogh (1994). We calculate the expected number of times each parameter is used to explain our input sequences, and compare this with the expected number of times each parameter should be used (i.e. positive examples should always take the upper paths in Figure 2, and negative examples should always take the lower path). We iteratively update our parameters according to:

$$\Theta_k^{+1} \leftarrow N(\Theta_k' + \eta(m_k - n_k)), \quad (2)$$

where  $\Theta_k'$  is the current value for the  $k$ -th parameter,  $m_k$  is the expected number of times it is used in correct paths to generate



**Fig. 4.** (a) A CRM logical structure. (b) Possible binding site locations on a DNA sequence  $x$ . (c) A dynamic programming matrix  $\alpha$ , where  $\alpha(A,i)$  represents the likelihood of sequence  $x$  from location  $i$  to  $L$  when site  $A$  occurs at location  $i$ . (d) A probability distribution over the locations of binding sites  $A$  and  $B$ , respectively. These probabilities tend to be extreme and high probabilities are sparsely distributed, which allows us to skip computation on all but the most likely values.

the training sequences,  $n_k$  is the expected number of times it is used in all paths,  $N$  is a normalization constant, and  $\eta$  is the learning rate, which can be adjusted dynamically so that the parameters do not fall below zero.

After we update model parameters, we do the appropriate normalization and smoothing. For most of the parameters, smoothing is done with pseudocounts. For our distance distributions, we smooth each histogram with a Gaussian-shaped kernel with standard deviation  $1/\sqrt{n}$  for a sample size of  $n$  (John and Langley, 1995).

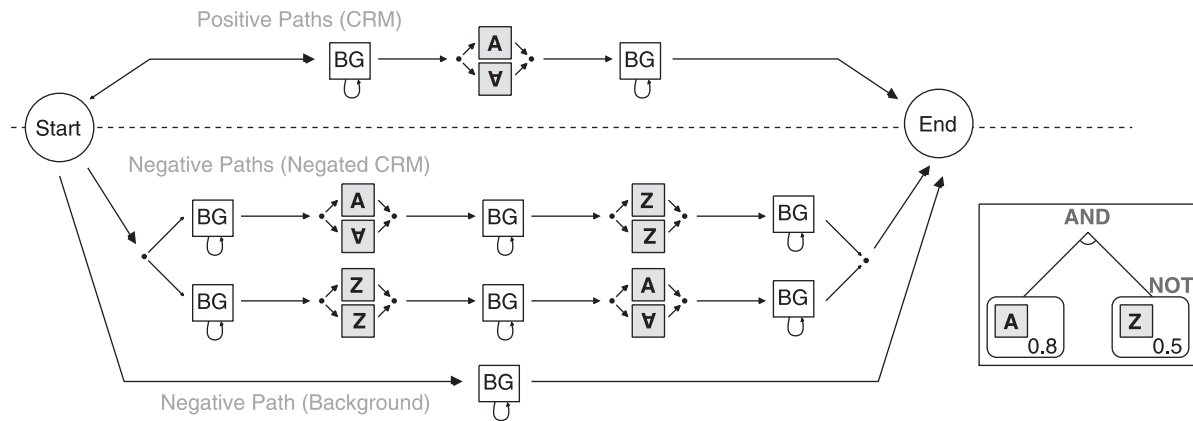
### 2.4 Efficient computation

To classify a sequence,  $x$ , we calculate the probability that  $x$  takes a positive path through the HMM. This is given by

$$P(c_x = \text{pos} | x, \Theta, V_x) = \frac{\sum_{\pi \in \Pi_{\text{pos}}} P(x | \pi) P(\pi | \Theta, V_x)}{\sum_{\pi \in \Pi_{\text{all}}} P(x | \pi) P(\pi | \Theta, V_x)}, \quad (3)$$

where  $\Pi_{\text{pos}}$  represents the set of positive paths through the HMM, and  $\Pi_{\text{all}}$  represents the set of all possible paths.

To understand how we compute (3), consider the example shown in Figure 4. Here, our model has two binding sites,  $A$  and  $B$ , as shown in Figure 4a. We consider each possible order separately, so assume the binding site order is fixed, with  $A$  upstream of  $B$ . Figure 4b. shows two possible locations for  $A$  and  $B$  on sequence  $x$ , which is of length  $L$ . To compute (3), we use a dynamic programming (DP) algorithm.  $\alpha$  is a DP matrix, shown in Figure 4c. Each entry in the matrix,  $\alpha_{(s,l)}$  represents the likelihood of the subsequence of  $x$ , from location  $l$  to  $L$ , given that binding site  $S$  occurs at location  $l$ . The iterative update step in the DP algorithm for



**Fig. 5.** An HMM with three groups of paths representing a CRM structure with a negated binding site (inset). A negative example sequence should take either the background path, or the paths that include the negated site.

**Table 1.** Results of finding CRMs in 25 yeast datasets from Lee *et al.* (2002)

Classification Dataset	Margin	<i>P</i> -value	Classification Dataset	Margin	<i>P</i> -value
GAT3, PDR1	0.765	<1.0e-5	GAT3, RGM1	0.467	<b>1.9e-3</b>
FHL1, RAP1	0.756	<1.0e-5	FHL1, YAP5	0.450	<b>2.2e-4</b>
GAT3, YAP5	0.691	<1.0e-5	MBP1, SWI4	0.440	<b>2.0e-5</b>
FKH2, SWI4	0.610	<1.0e-5	SWI4, SWI6	0.429	<b>1.0e-5</b>
NDD1, SWI4	0.603	<1.0e-5	MCM1, NDD1	0.424	<b>1.8e-4</b>
RAP1, YAP5	0.591	<1.0e-5	SKN7, SWI4	0.395	<b>2.4e-3</b>
FKH2, MBP1	0.580	<1.0e-5	FKH2, NDD1	0.350	<b>2.5e-4</b>
MBP1, SWI6	0.578	<1.0e-5	NRG1, YAP6	0.323	<b>5.8e-3</b>
MBP1, NDD1	0.570	<1.0e-5	GAL4, YAP5	0.313	0.053
FKH2, MCM1	0.540	<b>1.0e-5</b>	CIN5, NRG1	0.276	0.079
PDR1, YAP5	0.529	<1.0e-5	PHD1, YAP6	0.190	0.22
ACE2, SWI5	0.509	<b>3e-05</b>	CIN5, YAP6	0.160	0.21
RGM1, YAP5	0.467	<b>5.8e-4</b>			

Classification margins above the level of statistical significance ( $P$ -value < 0.01), which varies by dataset size, are shown in bold.

our two sites,  $A$  and  $B$ , with  $A$  at location  $i$ , is

$$\alpha_{(A,i)} = \sum_{j=i+w_A}^{L-w_B} \alpha_{(B,j)} \times P(x_{i..i+w_A} | \Theta_A) \times P(x_{i+w_A..j} | BG) \times P(j-i | \Theta_{\text{dist}(A,B)}), \quad (4)$$

where  $x_{i..j}$  represents the subsequence of  $x$  from  $i$  to  $j$ ,  $w_A$  and  $w_B$  are the widths of motifs  $A$  and  $B$ ,  $\Theta_A$  is the PWM for motif  $A$ ,  $BG$  is the background distribution and  $\Theta_{\text{dist}(A,B)}$  is the probability distribution in  $\Theta$  over the distance between  $A$  and  $B$ .

The run-time complexity of this calculation is  $O(L^2)$  (for two or more binding sites), which is impractical for long DNA sequences such as mammalian promoters. To make this computation tractable, we take advantage of one key insight: a large proportion of the sequence likelihood often depends on a small proportion of the combinations of binding site locations (i.e. where the sequence DNA actually matches the PWM distributions). Therefore, returning to our example in Figure 4, we first scan the sequence for the most likely locations for  $A$  and  $B$  (shown in Figure 4d), and consider those in order of decreasing likelihood, until we have considered

**Table 2.** Descriptions and classification margins of four datasets we use to test the effectiveness of our representation’s logical structure and spatial aspects ( $p$ -value < 0.01 shown in bold.)

Classification Dataset	Margin	<i>P</i> -value
Yeast ESR induced: 270 <i>S. cerevisiae</i> genes induced under ESR (5) (1000 negatives)	0.305	<1.0e-5
Yeast ESR PAC/RRPE cluster: 428 <i>S. cerevisiae</i> genes repressed under ESR. Promoters contain the PAC and RRPE elements (5) (1000 negatives)	0.338	<1.0e-5
Yeast ESR ribosomal proteins: 121 <i>S. cerevisiae</i> ribosomal protein genes repressed under ESR (5) (1000 negatives)	0.495	<1.0e-5
Fly gap system: 8 genes in the <i>D. melanogaster</i> gap system (13) (100 negatives)	0.730	<b>8.5e-5</b>

enough, e.g. 95% of the probability over all possible locations of both  $A$  and  $B$ . Figure 4c shows which cells (they are crossed out) in the DP matrix  $\alpha$  that we ignore because they contribute very little to the sequence probability. In practice, this decreases run time substantially and speeds up parameter convergence. It is a necessary step in learning CRMs from the long promoter sequences we wish to consider.

We calculate the expected number of times a parameter is used in (2) with a similar calculation, except that we need an additional DP matrix for the upstream subsequences. The expected number of times a parameter is used is proportional to the sum of the sequence likelihood over the paths that use it.

## 2.5 Negated binding sites

Recall that our model structure allows for negated binding sites (NOT logic). In these cases, there are three groups of paths through the corresponding HMM, as shown in Figure 5. The correct paths for positive examples are still the upper paths, corresponding

to the CRM model with negated binding sites removed. However, the ‘correct’ paths for the negative examples are divided among the lower (background) path, and the ‘middle’ paths (with negated sites), proportional to the likelihood that the example takes each path.

### 3 RESULTS

We wish to test whether or not our approach is able to learn accurate CRM models. To this end, we run our approach on several datasets, using cross-validation to measure the predictive accuracy of our learned models.

For the datasets that we consider, we train a 5th-order HMM on promoter sequences in the same genome to use as the background distribution.

As a metric for scoring learned models during our logical structure search, we use a statistic called  $F1$ . Given a trained model, we probabilistically estimate how many positive examples were generated by positive paths through our model (true positives,  $tp$ ), and through negative paths (false negatives,  $fn$ ), and how many negative examples were generated by positive paths (false positives,  $fp$ ). Precision is given by  $P = tp / (tp + fp)$ . Recall is given by  $R = tp / (tp + fn)$ .  $F1$  is the harmonic mean of precision and recall, and is given by  $F1 = 2PR / (P + R)$ .

#### 3.1 Evaluating predictive accuracy

In order to test our algorithm’s effectiveness in identifying CRMs, we compare our approach to that of Segal and Sharan (2004) on the same datasets. We recreated 25 yeast datasets where each gene in a given set has evidence ( $P$ -value  $< 0.001$ ) from the genome-wide analysis of Lee *et al.* (2002) that two particular proteins bind somewhere in its upstream region. For each dataset, we use 500 bp promoter sequences, and choose 100 yeast promoter sequences at random to use as negative examples.

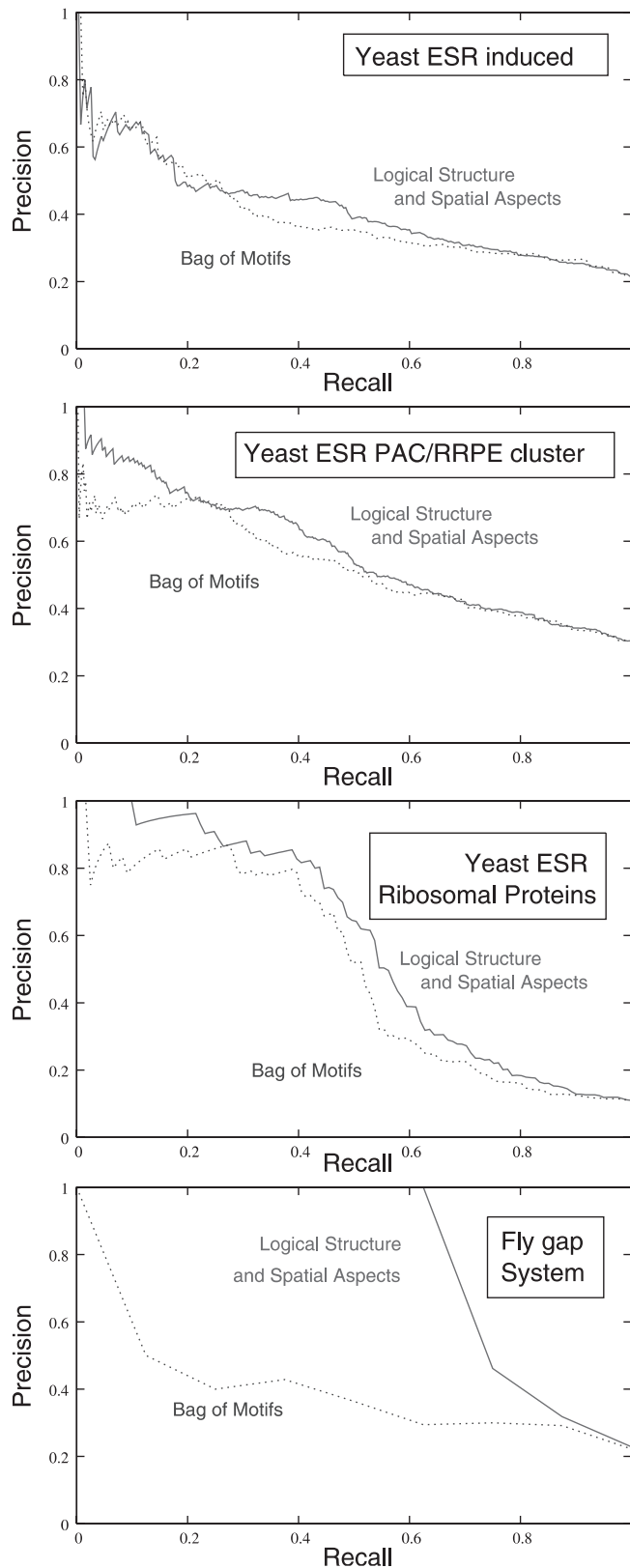
To show that the predictions of our learned models on held-aside data are more accurate than could be obtained by chance, we compute a classification margin (following Segal and Sharan) which is the largest difference between the true positive rate and the false positive rate as a threshold is varied on what is called a positive example. If there is  $< 1\%$  chance that a randomly-labeled test set with the same cardinality of positive and negative examples would have the classification margin of one of our test sets (or a higher one), then we consider this result statistically significant.

Our results are shown in Table 1. We find significant results in 21 of 25 datasets, compared to 12 of 25 found by the approach developed by Segal and Sharan.

Recall that we train our models using a discriminative approach. Our experiments show that our learner is more accurate than models learned using a standard, generative training approach. Of the 30 datasets mentioned in this section, the discriminative method finds more accurate models for 20 of them, especially on the yeast datasets described in the next section.

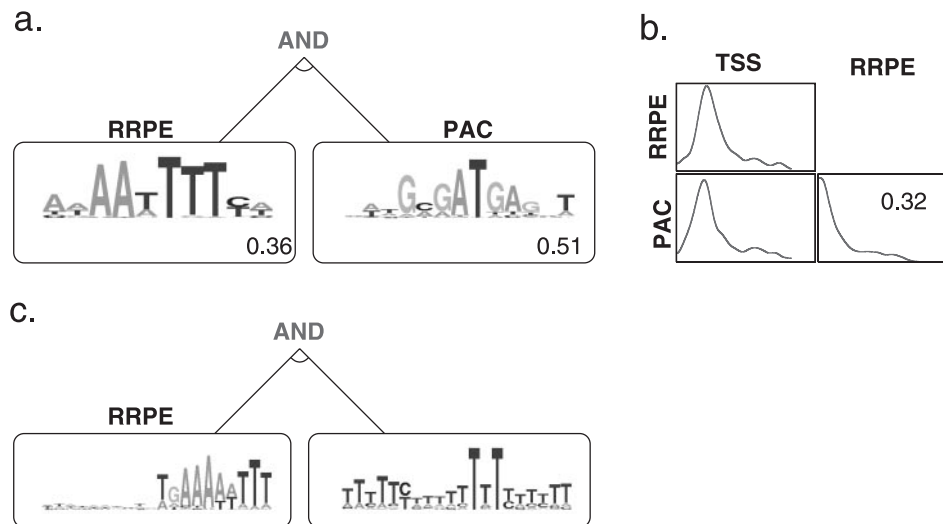
#### 3.2 Evaluating the effectiveness of logical and spatial aspects

In order to evaluate whether the logical structure and spatial aspects of our representation improve the ability of our learner to recover CRMs, we compare our approach to a restricted version wherein we do not allow logical structure beyond AND, and all our spatial



**Fig. 6.** Precision–recall curves from four datasets described in Table 2. The accuracy of our models dominates that of the bag-of-motifs approach over almost all of the recall space in these datasets.





**Fig. 7.** (a, b) The hypothesis CRM model learned by our algorithm for the yeast PAC/RRPE dataset (Table 2). Both the PAC element (consensus sequence GCGATGAG) and the RRPE element (consensus sequence AAAAAwTTTT) appear as overrepresented in our learned PWMs. (c) The model learned by the bag-of-motifs approach on the same dataset.

probabilities are fixed by a uniform distribution. That is, the model space of this restricted version is simply a conjunction of motifs which may appear in any order, in any location, and so we refer to it as the ‘bag-of-motifs’ approach. The classification margin is higher using our approach than using the bag-of-motifs approach on 16 of the 25 Lee *et al.* datasets described above.

We test our approach on four additional datasets from both yeast and fly, for which we obtain promoter sequences from genes known to be co-regulated. Table 2 describes these datasets and includes a classification margin and  $P$ -value showing that we find statistically significant CRMs in all four datasets. Since there is a large discrepancy between the number of positive and negative examples in these datasets, we create precision–recall (PR) curves, which show the tradeoff between precision and recall over classification thresholds. These results are shown in Figure 6.

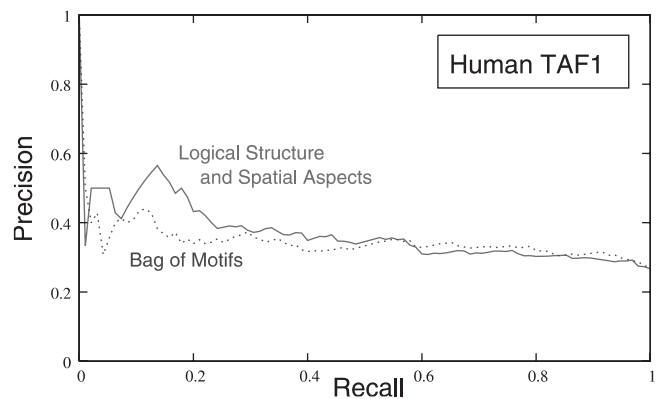
In each case, the PR curve for our model dominates the PR curve for the bag-of-motifs model over all or almost all of the recall space.

The yeast ESR PAC/RRPE genes described in Table 2 contain two known elements in their upstream regions, the PAC element (consensus sequence GCGATGAG) and the RRPE element (consensus sequence AAAAAwTTTT). Figure 7 shows the hypothesis CRM model learned by our approach (Figure 7a and b) and that of the bag-of-motifs approach (Figure 7c), when trained on the entire dataset.

The PWMs recovered by our algorithm are shown in Figure 7a as sequence logos (Crooks *et al.*, 2004), which show a high amount of overlap with the known consensus sequences. The bag-of-motifs approach did not recover the PAC element. This example illustrates how the inclusion of spatial preferences in the representation can aid the learner in finding better motif models. Moreover, the inclusion of these aspects leads to more accurate classifications even when the ‘right’ motifs have been learned.

### 3.3 CRMs in human

In order to determine whether or not our approach can be effective in finding CRMs in DNA sequences in more complex organisms



**Fig. 8.** The precision–recall curve for our approach compared with the bag-of-motifs approach on human DNA sequences.

than yeast and fly, we test our approach on several human promoter sequences annotated with GO term 3677 (DNA binding proteins).

This set consists of 95 positive examples of 4000 bp regions that have evidence of being bound by a transcription factor called TAF1, and 284 negative examples with evidence of not having a TAF1 binding site (these data were obtained from the Thomson lab at the University of Wisconsin; unpublished data).

The classification margin we obtain from this dataset is 0.220 ( $P$ -value =  $8.9e - 4$ ). The comparison of precision and recall with the bag-of-motifs approach is shown in Figure 8.

## 4 CONCLUSION

We have presented a probabilistic learning algorithm which is capable of learning multiple motifs and rich representations of logical and spatial relationships among them simultaneously. Our models can be thought of as generalized HMMs, but they are specifically designed to represent aspects of CRMs. We learn their structure by

searching for the logical structure of the underlying CRM, and our representation is compact because of extensive parameter sharing. We have also presented a learning algorithm to train these HMMs, which uses a heuristic approach to make it efficient enough to learn from mammalian sequence data.

We have shown that our motif learner performs better than a current state-of-the-art approach on the 25 yeast datasets from Lee *et al.* and we have shown that learning information about the logical structure and spatial aspects of a CRM helps our learner find better models on five datasets, as measured by predictive accuracy.

## ACKNOWLEDGEMENTS

The authors would like to thank Audrey Gasch and James Thomson and their groups for help with the datasets. This research was supported in part by NIH/NLM training grant 5T15LM005359 and NSF grant IIS-0093016.

## REFERENCES

- Aerts,S. *et al.* (2003) Computational detection of cis-regulatory modules. *Bioinformatics*, **19**, 5–14.
- Beer,M.A. and Tavazoie,S. (2004) Predicting gene expression from sequence. *Cell*, **117**, 185–198.
- Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.
- Crooks,G.E. (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–1190.
- John,G.H. and Langley,P. (1995) Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345. Morgan Kaufmann.
- Keleş,S. *et al.* (2004) Regulatory motif finding by logic regression. *Bioinformatics*, **20**, 2799–2811.
- Krogh,A. (1994) Hidden Markov models for labeled sequences. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pp. 140–144. IEEE Computer Society Press.
- Lee,T.I. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.
- Mitchell,T.M. (1997) *Machine Learning*. McGraw-Hill, New York.
- Noble,W.S. *et al.* (2005) Predicting the *in vivo* signature of human gene regulatory sequences. *Bioinformatics*, **21**, i338–i343.
- Segal,E. and Sharan,R. (2004) A discriminative model for identifying spatial cis-regulatory modules. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology (RECOMB)*, pp. 141–149. San Diego, California, USA. ACM Press.
- Sinha,S. *et al.* (2003) A probabilistic method to detect regulatory modules. *Bioinformatics*, **19**, 292–301.
- Zhou,Q. and Wong,W.H. (2004) CisModule: *De novo* discovery of cis-regulatory modules by hierarchical mixture modeling. *Proc. Natl Acad. Sci. USA*, **101**, 12114–12119.