Bioinformatics Advance Access published June 5, 2008

Sequence Analysis

# Modeling Promoter Grammars with Evolving Hidden Markov Models

Kyoung-Jae Won[1,#], Albin Sandelin[1], Troels Torben Marstrand[1], Anders Krogh[1,*]

[1] The Bioinformatics Centre, Department of Biology & Biotech Research and Innovation Centre, University of Copenhagen, Ole Maaloes Vej 5, 2200 Copenhagen N, Denmark

# Present address: Dept of Chemistry & Biochemistry, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0359, USA

Associate Editor: Dr. Alex Bateman

**ABSTRACT**

**Motivation:** Describing and modeling biological features of eukaryotic promoters remains an important and challenging problem within computational biology. The promoters of higher eukaryotes in particular display a wide variation in regulatory features, which are difficult to model. Often several factors are involved in the regulation of a set of co-regulated genes. If so, promoters can be modeled with connected regulatory features, where the network of connections is characteristic for a particular mode of regulation.

**Results:** With the goal of automatically deciphering such regulatory structures, we present a method that iteratively evolves an ensemble of regulatory grammars using a Hidden Markov Model (HMM) architecture composed of interconnected blocks representing transcription factor binding sites and background regions of promoter sequences. The ensemble approach reduces the risk of over-fitting and generally improves performance. We apply this method to identify transcription factor binding sites and to classify promoters preferentially expressed in macrophages, where it outperforms other methods due to the increased predictive power given by the grammar.

**Availability:** The software and the data sets are available from http://modem.ucsd.edu/won/eHMM.tar.gz

**Contact:** krogh@binf.ku.dk

## 1 INTRODUCTION

One of the fundamental challenges in computational biology is to decipher the signals underlying transcriptional regulation (Stormo, 2000; Wasserman and Sandelin, 2004). The goal of this is twofold: to minimize the number of experiments necessary in the laboratory, but also to understand the general mechanism underlying the precise selection of expressed genomic loci. Transcription of a typical gene by RNA Polymerase II is directed by DNA sequence signals, which are bound by specific proteins: transcription factors (TFs). These transcription factor binding sites (TFBSs) are often located in the region around the transcription initiation site (Smale and Kadonaga, 2003). The TFBSs for a given TF usually show a constrained pattern of nucleotides, which can be represented by a position-specific scoring matrix (PSSM) (Stormo, 2000; Wasserman and Sandelin, 2004). While such PSSMs are adequate predictors of sites bound in vitro, the information content in the model is too small to make meaningful predictions at genomic scales. The binding preference of a TF alone is not sufficient to find its cognate functional sites. One of the principal methods used to solve this problem is to find combinations of sites. Such combinations of TFBS are known as *cis*-regulatory modules, which can direct tissue-specific expression of genes (Stormo, 2000; Wasserman and Fickett, 1998).

In higher eukaryotes, it is challenging to model the interaction between TFs and TFBSs, as the location, composition and number of TFBSs varies greatly even in genes having similar expression patterns. A bottleneck in this type of analysis is the lack of data: there are only a handful of gene sets where the number of experimentally defined sites is sufficient for direct training of a predictive model. On the other hand, genome-wide data on both promoter locations (Carninci, et al., 2006) and expression patterns (Su, et al., 2002) are available. Thus, a commonly occurring situation in experimental biology is that a set of genes are found to be co-expressed, leading to the hypothesis that they are co-regulated or at least share some regulatory features. In some of these cases, there are experimental indications on what type of features those might be (for instance, some particular TFs might be suspected to be involved in the regulation of most genes within the set). Many algorithms aimed at analyzing data originating from this type of situation have been presented. Most have focused on identifying regions in which predicted sites co-occur (Berman, et al., 2002; Markstein, et al., 2002; Rebeiz, et al., 2002). Others have suggested probabilistic models (Crowley, et al., 1997; Frith, et al., 2001; Frith, et al., 2003; Frith, et al., 2002; Rajewsky, et al., 2002). Of particular interest are Cister and COMET, designed using two parallel strings of hidden Markov model (HMM) states to represent forward and backward reading of a PSSM (Frith, et al., 2001; Frith, et al., 2002). Background states are used to model spacing between PSSMs. Comet calculates E-values considering the score from the HMM and the gap between the putative TFs. A more recent program, Cluster-Buster, employed a quadratic-time algorithm to find clusters of pre-specified motifs in nucleotide sequences (Frith, et al., 2003). Similarly, Stubb is a program that uses correlation between motifs to model the coordinated TFBSs and also incorporate phylogenetic information (Sinha, et al., 2003).

Given a reasonable set of PSSMs as input, the applications using probabilistic models have adequate performance in the sense that they can successfully locate known clusters of motifs.

However, these approaches do not consider the internal structure of the co-occurring TFBSs, just the proximity or co-localization of motifs. This is the motivation for our study: we focus on how to model the promoters of such a group of genes both to gain understanding of the mechanism of regulation and to be able to search for other genes that are regulated the same way. This can be viewed as an extension of the above approaches – not only do we want to find *cis*-regulatory modules; but to model the "grammar" of the promoters, by modeling both the binding sites, the order in which they appear, and the regions in between them. Thus, for a set of genes which are hypothesized to have similar promoter structures, we make a model by combining known PSSMs with models of the regions in between them: in effect, an HMM architecture based on smaller blocks which in turn also are HMMs. We use Genetic Algorithms (GAs) to automatically optimize the HMM architecture, starting from a simple network. This is similar to another evolutionary method for motif discovery, EMCMODULE (Gupta and Liu, 2005). Using an evolutionary Monte Carlo method, EMCMODULE updates the motifs and their locations in the sequences in each genetic cycle. It also has some similarity to Stubb (Sinha, et al., 2003), which uses inter-correlations of motifs by constructing a 'history-conscious HMM', where a previous non-background motif is remembered.

In this work, we search for multiple models, each of which represents one aspect of the underlying grammar. The evolving strategy searches for possible motif grammars, while transforming the internal representation of the context. We use an evolutionary algorithm to optimize an HMM structure (Won, et al., 2007). The structure learning method has been successfully applied to prediction of protein secondary structure (Won, et al., 2007) and prokaryotic promoters (Won, et al., 2006) . The method we present here differs from the previous work in that we use a special block designed to model a PSSM. We also present a way of designing a promoter classifier using PSSM blocks.

Below, we show that our method can find dense clusters of sites with performance equal to or better than other methods, and can successfully reconstruct known regulatory grammars at the same time. We also show that our method can be used as a classifier, where the modeled grammar gives additional predictive power compared to other approaches.

## 2 METHODS

### 2.1 Evolving an HMM

We present an evolutionary strategy that evolves the structure of an HMM. The search space is restricted to interconnections of static sub-structures called blocks, which correspond to regulatory features or spacers (Won, et al., 2006). Such evolved HMMs we call e-HMMs. Four types of blocks were used (Figure 1): linear, self-loop, forward-jump blocks and zero blocks. A linear block consists of N states, where each state only has a transition to the next state. These linear blocks are used to model sites with a certain length (similar to a PSSM). Self-loop blocks are linear blocks in which each state has an additional transition to itself. They are used to represent a background distribution and model the length distribution between other blocks. A forward-jump block is a linear block where the first state is also connected to the last M states (with 1≤M<N) to model spacers of varying lengths up to a (small) maximum length. Zero blocks are empty blocks with no states, which are just place holders in the model structure and may change to another type of block at some point (through mutation or cross-over). In addition to the four types of blocks suggested in our previous work, we use a type that represents forward and backward reading of a PSSM for to model TFBSs (Figure 1e), since functional TFBS generally can occur on both strands. Each end of the PSSM block is a silent state that does not emit any symbols, but has transitions to other states and these transitions can capture possible directional preferences of the binding sites. Inside a PSSM block the emission probabilities are set according to the base frequencies of a known PSSM with a pseudo count of 1, and these probabilities are fixed. The number of states in a PSSM block is $2q + 2$, where $q$ is the length of the corresponding PSSM.

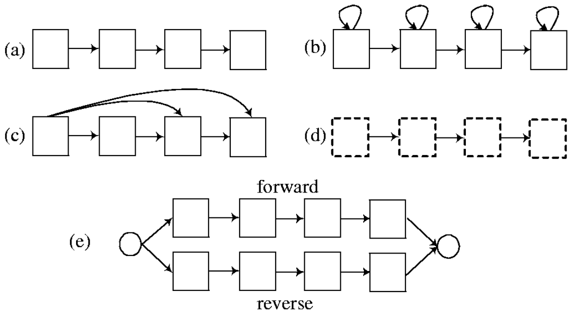

**Fig. 1.** HMM block types used. (a) linear block (b) self-loop block (c) forward-jump block (d) zero block (e) PSSM block.

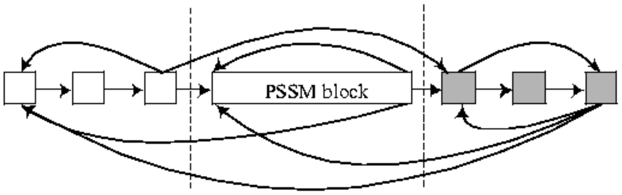

**Fig. 2.** Example of an HMM architecture. The architecture is composed of two background blocks and one PSSM block. All the blocks are fully connected to each other. The blocks are divided by dotted lines

All block types except the PSSM blocks are called background blocks as they are used to model background DNA sequence or spacing between TFBSs.

An HMM consists of a fixed (preset) number of blocks, some of which may be zero blocks. Initially, block types are randomly allocated and the blocks are fully connected to form a complete HMM. In this context, fully connected means that the end state of each (non-zero) block is connected to the start states of all other blocks and itself. An example of an HMM composed of three blocks is shown in Figure 2. States in background blocks are tied within blocks, meaning that their emission probabilities of the states inside a block are identical at all times. Thus, in the example depicted in Figure 2, the first three states are tied, and so are the last three states, but emission probabilities of the states in the first and the last block are different.

The method requires a set of PSSMs that are believed to be overrepresented in the input set of sequences. The aim of the method is then to find an optimal HMM architecture containing these PSSM blocks as well as evolving spacer blocks that explains the sequences. To do this, the genetic algorithm starts from a number of randomly chosen blocks to form a population. Inside the genetic cycle, genetic operators select members of the population (called parents) and evolve them to produce new members (called children). Children and parents are evaluated using a fitness function based on the likelihood. According to the fitness function, the selection procedure selects a number of members in the population for the next genetic cycle. Genetic operators (crossover, mutation) are applied to evolve the structure while retaining the properties of blocks. Crossover operations swap a number of blocks in two parents to create two children. Mutations change the number of transitions or states in the children. In each cycle, the best member in the population each cycle is stored. See Supplementary text for more detail.

## 2.2 Genetic operators and training of e-HMMs

Genetic algorithms evolve a population of solutions with genetic operators. Two genetic operators are used: crossover and mutation (Won, et al., 2006). In crossover, two members are chosen through the selection procedure described below. A number of blocks are randomly selected and swapped to create two children. The number of blocks is constant, but the number of the states of an HMM can be changed. Fig S1 illustrates an example of the crossover scheme. Mutations can either change the block type or the state structure of a block. In a block-type mutation, another type is selected at random. Mutations to the state structure can happen inside any block of the HMM except the PSSM blocks (Fig S2). They change the number of states and the number of transitions in a block by randomly inserting or deleting a state. By applying these operators, the e-HMM evolves the model. To obtain suitable HMM architectures we tested various numbers of blocks between 25 and 40. Table S1 shows parameters used in the simulation. We have used a hybrid GA with traditional genetic operators to explore the space of HMM topologies in combination with Baum-Welch optimization of the transition and emission probabilities. After a number of iterations, most of the initial transitions converge to zero. The remaining transitions decide the grammar of the evolved HMM. The log likelihood of a model also tells us how well it fits the data. Given an HMM, we calculated the fitness value using the Akaike Information Criterion (AIC) (Akaike, 1973), which balances the likelihood and the model complexity (the number of parameters). The fitness value is

$$E_\mu = \frac{1}{-\sum_i \left\{-2\log(P(\mathbf{x}_i \mid \Theta_\mu)) + 2\lambda f_\mu\right\}/l_i} \qquad (1)$$

where $l_i$ is the length of a sequence $\boldsymbol{x}_i$ and $\mu$ labels the different HMMs (with parameters $\boldsymbol{\Theta}_\mu$ ) of the population. The number of free parameters in the HMM is called $f_\mu$, and the parameter $\lambda$ balances the likelihood and the complexity of the HMM.

A member of the population is selected with the Boltzmann probability

$$F_\mu = \frac{m_\mu}{\sum_{v=1}^{N} m_v}, \quad m_\mu = e^{sE_\mu/\sigma} \qquad (2)$$

where $\sigma$ is the standard deviation of the fitness in the population and $s$ is a constant that controls the strength of the selection. In the work reported here, we used a value of $s$ equal to 0.3 and $\lambda$ equal to 0.5.

## 2.3 Parameter training using PSSM scores

The evolved model is trained again considering the PSSM matches in a sequence and the distances to other matches. It is likely that a putative binding site with high matrix score and located close to other binding sites is a true binding site. To train with this additional biological information, we adopted the method for including database matches previously used for gene detection in *Drosophila* (Krogh, 2000). We assigned a probability distribution over labels to each base in the sequence based on the PSSM score and the distance to other candidate TFBSs. A label is associated with a TF or background. An HMM state emits only a single label. The training algorithm with labels calculates a path where a state label matches with a sequence label. By assigning multiple labels to a sequence, multiple PSSM blocks or background can have a path through each base in the sequence. Firstly, putative binding sites are obtained by considering the PSSM score.

$$pssm(TF_i) = s_i = \log \frac{\prod_{k=1}^{W_x} P_k(x_k)}{\prod_{k=1}^{W_x} P^b(x_k)}, \qquad (3)$$

where $W_x$ is the width of a PSSM and $P_k\ (x_k)$ is the probability of observing nucleotide $x_k$ at position $k$ of the PSSM, and $P^b(x_k)$ is the probability of observing nucleotide $x_k$ according to the background distribution. Non-binding sites are the sites with the PSSM score less than a cut-off. Secondly, the label probabilities are set for each base in the sequence. There is a distinct label for each PSSM and one for background, and a letter has one of those labels with some probability. For example, if a region of a sequence has a PSSM score $s_1$ for transcription factor TF1 and $s_2$ for TF2 that are the only ones that score above the threshold, then $pssm(TF1)= s_1{\cdot}z$, $pssm(TF2)= s_2{\cdot}z$, $pssm(TF3) = pssm(TF4) = \ldots = pssm(background) = s_\alpha \cdot z$, where $z$ is a normalization factor and $s_\alpha$ is a pseudo count. Pseudo counts are used in order to avoid label probabilities of 0. Table S2 gives the value before normalization assigned to each label based on its PSSM score. The normalized distribution is scaled using a confidence. Motifs are usually found clustered in real data. If any other putative binding sites are found within a certain distance, a confidence of 1 is used, and otherwise 0.1. We observed the performance with various distances $D$ along with the decoding method. The scaled probability of the $k$th label at position $l$ is

$$p'_l[k] = \text{(1-confidence)/(number of TFs+1)} + \text{confidence} \times pssm_l[k]. \qquad (4)$$

If the confidence is 0 it becomes a uniform distribution of labels. The probability of yielding a sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$ along a path $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_L)$ and a label $y = (y_1, y_2, \ldots, y_L)$ in an HMM is

$$P(\mathbf{x}, \mathbf{y}, \pi) = \prod_{l=1}^{L} a_{\pi_{l-1}\pi_l} e_{\pi_l}(x_l)\delta(y_l = c(\pi_l))p'_l(y_l) \qquad (5)$$

where $a_{ij}$ is a probability of making transition from state $i$ to state $j$, and $e_i(x_l)$ a probability of emitting a symbol $x_l$ in a state $l$ . $c(\pi_l)$ is a label in the state $\pi_l$ . $\delta$ is the Kronecker delta function. It is 1 if $y_l = c(\pi_l)$ , and 0, otherwise. These probabilities are multiplied along a path, so the probability of not using a path with high label probabilities is heavily penalized. See Supplementary text for more details.

## 2.4 Decoding e-HMMS and interpreting e-HMM ensembles

In each iteration of the genetic algorithm, new models are estimated. Many of these models are equally good, and choosing one "best" model in the end rarely yields the best results. Therefore we use an ensemble of models to analyze the data, which is a well-known method for limiting over-fitting (Riis and Krogh, 1996). Depending on the problem, the way of decoding the e-HMM and interpreting an ensemble of e-HMMs differs. To find putative binding sites we used posterior state probabilities to calculate the most probable states for the given sequences (Durbin, et al., 1999). For each position in a sequence, the state with the highest probability is chosen, and if it is a state in a PSSM block, the position is predicted to be a binding site for the corresponding transcription factor. Associated with each site is a score of the sequence given a PSSM (3). If a predicted site has a score below a chosen cut-off, it is discarded. Among the remained sites we only counted predicted sites that clustered. Any two sites were regarded as clustered if the gap between them was smaller than a certain size (D). We checked the performance while varying D from 80 to infinity. When D is infinite, all predicted sites above the cut-off are kept. We add up the number of times each site is predicted in an iteration of the GA: . If a site is predicted as a specific motif 10 times in 15 iterations, the ensemble method gives a score of 10/15 to the site. Consequently, this method locates signals surviving for a long period of genetic permutations.

## 2.5 Constructing a classifier

To design a classifier using e-HMMs we calculated the log-odds ratio

$$\log \frac{P(\mathbf{x} \mid \Theta^{+})}{P(\mathbf{x} \mid \Theta^{-})} \qquad (6)$$

where $\boldsymbol{x}$ is the sequence and $\Theta$ is a set of HMM parameters. The positive model ($\Theta^{+}$) is an evolved e-HMM. To design the negative model ($\Theta^{-}$) we took all PSSM blocks out of the positive model and used only background blocks. In the example shown in Figure 2, the negative model becomes a 2-block HMM, without the PSSM block in the middle. The negative model is trained using the Baum-Welch algorithm with a set of negative sequences believed not to share the same grammar or sites as the positive set. For an ensemble of *N* models, we used the averaged log-odds ratios for classification:

$$\frac{1}{N}\sum_{i} \log \frac{P(\mathbf{x} \mid \Theta_{i}^{+})}{P(\mathbf{x} \mid \Theta_{i}^{-})} \qquad (7)$$

In this paper, we used a promoter set expressed in macrophage and conducted 3-fold cross validation.

## 2.6 Data set construction

### *2.6.1 Muscle-specific sequence set*

We used the muscle-specific promoter sequences as downloaded in the 24.nonaligned.pos.train.fa.tar.gz file at http://bayesweb.wadsworth.org/ gibbs/module/ and extracted a 1000-1500 nt region from each included sequence that contained all the annotated TFBS. The sequences are mostly 1000 nt long; in the few cases when this span did not cover all TFBS, a 1500 bp span was used

### *2.6.2 Macrophage-specific sequence set*

For the macrophage promoter set construction, we used CAGE-derived promoters. CAGE is a method for sequencing the first 20-21 nucleotides of full-length cDNAs. A unique strength of the method is nucleotide resolution TSS detection coupled with a data depth enabling measuring the tissue specificity of core promoters and individual TSSs. CAGE tags were sequenced and mapped to the mouse genome (assembly MM5) as described in (Carninci, et al., 2006). 11,567,973 CAGE tags were sequenced from more than 20 tissues, using 144 distinct CAGE libraries. Tag clusters (tags mapped to the genome that overlap with at least one nucleotide on the same strand) with more than 30 tags from the libraries studied were used. We focused our analysis on tags from libraries derived from lipo-poly-saccharide (LPS) induced bone marrow macrophages (Library IDs: CBV, CBW,CBX, CBY, CBZ, CCA, CCB,CDR, CDS, CDT, CDU, CDW, CDY) and used remaining tags for assessing how constrained expression was to macrophage regions.In total, 15717 tag clusters were analyzed. If a tag cluster had more than 60% of tags from the LPS-induced set (correcting for sample size), it was considered to be LPS-induced specific, otherwise the cluster was labeled negative. For each cluster, a sequence region of -300 to +50 in relation to the most used TSS in the cluster was extracted from the mm5 genome assembly for the testing. This resulted in a positive set of 503 LPS-induced promoters and 15314 negative promoters. We conducted a 3-fold cross-validation with this set of sequences.

# 3 RESULTS

## 3.1 E-HMM performance

For testing the method, we constructed three test sets aimed to assess the performance of our e-HMMs to both classify promoters and find motifs;

(1) Two artificial sets, where we first assess how well our method can find implanted motifs compared to other methods, and then whether it can rediscover the artificially constructed promoter grammar.

(2) A well-annotated experimental set of muscle-specific promoters, where we assess how well-known motif clusters can be found compared with two other methods, and present some insights into the underlying regulatory grammar in these promoters.

(3) An un-annotated set of core promoters preferentially expressed in macrophage cells involved in immunological response, where we test the ability to classify promoters with respect to tissue specificity.

### *3.1.1 Reconstruction of a known grammar*

We generated 120 artificial sequences using an HMM with 5 muscle specific PSSM(Wasserman and Fickett, 1998) blocks and some background blocks. This is equivalent to “knowing” the underlying grammar in these sequences. An important caveat with this study is that all the sequences were constructed with the same grammar, so there are no other grammars present. The generated sequences have 501 TFBSs (121 MYODs, 120 MEFs, 120 SRFs, 79 SP1s and 61 TEFs). For PSSMs we used V_$MYOD_Q6_01, V_$SRF_Q4, V_$MEF2_02, V_$TEF_Q6 and V_$SP1_Q6 PSSMs from the TRANSFAC database (Matys, et al., 2003).

Starting with 25 blocks, we evolved a population of HMMs using the artificial sequences and the same PSSMs without any prior information on the HMM that generated the sequences.

First, we assessed how well the e-HMMs predict locations of motifs. Figure 3 compares the result of the e-HMM, COMET, and Cluster-Buster. COMET and Cluster-Buster are TFBS predictors given the sequence and PSSMs. We ran COMET and Cluster-Buster while varying E-value and cluster threshold, respectively from 0 until it has a maximum number of predictions.

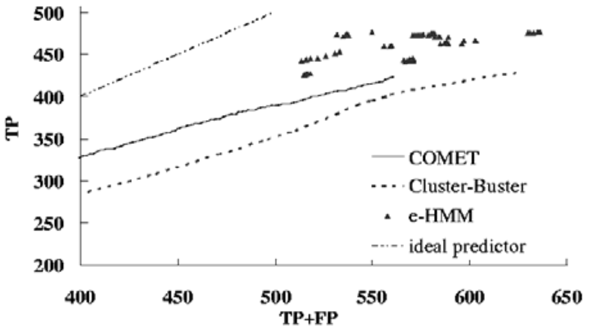

Fig. 3. Performance of e-HMMs as motif detectors on an artificial promoter set. True positives are plotted against the total number of predictions (true positive + false positives). For the e-HMMs, decoding of individual HMMs is used – the ensemble approach is not used.

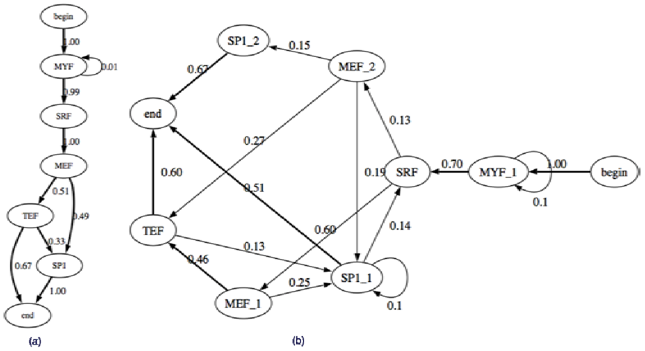

Fig. 4. Reconstruction of a known grammar. An artificial grammar (and corresponding sequences) using 5 known PSSMs was constructed using an HMM: the HMM graph is shown in a). Background states used in modeling gaps between PSSM blocks are not shown in this diagram. Panel b) shows the inferred grammar by applying an e-HMM on the sequences using the same PSSMs but no other prior knowledge. Only transitions with P> 0.1 are drawn. The full diagram is found in Fig. S5. Note that the most prominent edges in the original HMMs are rediscovered by the e-HMM. Blocks describing the SP1 and MEF TFBS are duplicated, although one of the MYF blocks is not connected to any other block (not shown).

We counted the number of correctly predicted TFBSs (TP) and incorrectly predicted TFBSs (FP) while changing the cutoff value of Cluster-Buster and COMET. As our algorithm is not deterministic, the performance of the e-HMMs fluctuates. Nevertheless, the e-HMMs perform substantially better than both Cluster-Buster and COMET on this set. Next, we investigated to what degree the evolved HMMs can reconstruct the original grammar (this is equivalent to the similarity of the HMM that constructed the artificial sequences). Figure 4a shows the HMM used for construction of the artificial sequences, while Figure 4b shows the corresponding remodeled directed graph acquired after decoding the sequence through one of the evolved HMMs. This particular model found 438 TPs among 607 predictions (TPs+FPs). The figure shows that the evolved HMM approximates the original (correct) grammar quite well. For example, the transitions from 'TEF' are modeled well with high transition probabilities to 'end' and 'SP1_1', and other transitions have $p < 0.1$. Thus, in this example of a "known" grammar, with no additional grammars present, our model is better at finding the locations of the grammar elements and can reconstruct the grammar at the same time. In supplementary text, we describe a more generalized test using a graph where all TFs are connected, and transitions are randomly assigned (Fig. S6, Table S4 and supplementary text); the correlation between assigned and predicted transition probabilities is generally successful

### *3.1.2 Analysis of an annotated set*

Above, we observed that the HMM method has the power to model the existing grammar of motifs. The grammar of real data is composed of many rules – either variants of the same grammar or multiple grammars that are different. To model this, rules that are evolved in each stage are collected and the most conserved ("stable") grammar during the artificial evolution are evaluated (see Methods).

To assess the performance with real data, we used the 48 human and mouse sequences (Wasserman, et al., 2000) containing 166 experimentally verified TFBSs from muscle-specific promoters. It is commonly assumed that these sequences share a transcriptional logic. For this simulation we used 5 TF models: MYF, MEF2, SRF, TEF and SP1 from the JASPAR database. As above, we compared the result of COMET, Cluster-Buster and our method to first see if the known TFBSs can be detected, and then assess the hypothesized underlying grammar. Figure 5 shows the correct predictions (TP) vs. the total number of predictions (TP+FP) of COMET, Cluster-Buster and e-HMMs. The performance of individual e-HMMs (Figure 5) is consistently better than Cluster-Buster and overall, the performance of individual e-HMMs is comparable to that of COMET. As discussed above, our algorithm can also construct an ensemble of e-HMMs to include prediction information from individual e-HMMs(see methods). The ensemble models perform better than individual e-HMMs in almost all instances, and is generally closer to the performance of a perfect predictor, although in the range of 60-150 total predictions (x axis), the COMET method has comparable performance (See Figure S3). When the number of predictions is larger (>150), the ensemble method outperformed the other methods. Figure S4 compares the receiver operator characteristic (ROC) curve of the three methods: the eHMM method performed better than the other methods.

For clarity, the above test only measures if the known TFBS are found, and not the underlying grammar. The HMM structure is only used to weight sites higher if they cluster – we explore the effect of changing these settings in the Supplementary text. As a summary, the e-HMMs are generally comparable or better than COMET and Cluster-Buster for finding the known sites.

However, this is not the main attraction with the model: what sets e-HMMs apart from the other methods is that we can assess the grammar to form a hypothesis of the underlying biology. Table S3 shows strong edges between motifs in two evolved HMMs (resulting from iteration 100 and 120, respectively). We counted the number of grammars of the HMM at the $100^{th}$ and $120^{th}$ iteration and found 78 grammar and 72 grammar respectively after decoding. Among the grammar 'MYF→MEF' is well preserved. 'SP1' is not frequently found in the HMM at 100th iteration, but new rules from 'SP1' are emerged during the evolution, as new edges from SP1 are present in the HMM from the 120th iteration. Conversely, the 'TEF→MEF' edge decreased significantly.

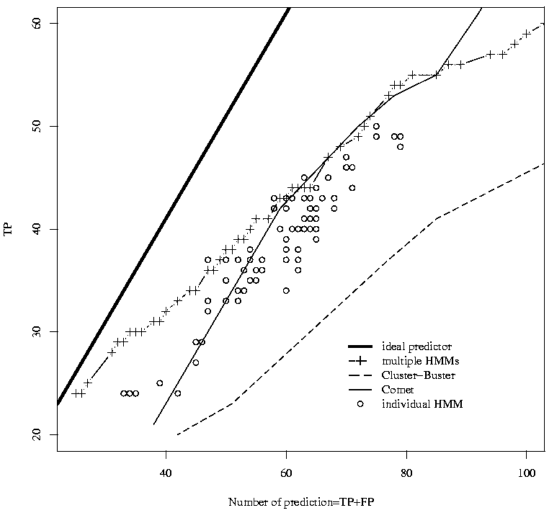

Fig. 5. Performance of e-HMMs as motif detectors on an annotated promoter set. True positives are plotted against the total number of predictions (true positive + false positives). The performance of e-HMM is compared with Comet and Cluster-buster. An extended variant of this plot is shown in figure S3 and figure S4.

### 3.1.3 Classification of macrophage-specific core promoters

Most interesting data sets are not as thoroughly analyzed and annotated as the muscle set that we analyze above. In genomics, we often face the situation where some promoters are shown to be active under some stimuli, and others are not. In general, in these situations we have no knowledge about the functional sites in these promoters, although often some hypothesis exists regarding the identity of the transcription factors involved (so, we might have the identity of the factors but not their sites nor the regulatory grammar). This type of situation is a case where we can use e-HMMs as classifiers, if it can find grammars that distinguish two sets of promoters from one another. To test this, we choose a large novel biological promoter set where we have some indications of which TFs that might be determinants of tissue-specific expression, but not the details of its mode of regulation. Specifically, we extracted promoters preferentially expressed in mouse macrophages induced by lipopolysaccharide (LPS), as measured by Cap Analysis of Gene Expression (CAGE) data presented in (Carninci, et al., 2006 )(See Methods). LPS treatment provokes a drastic expression response similar to that of *in-vivo* immunological response (Nilsson, et al., 2006). To train the e-HMMs, we selected PSSMs among the TFs presented in (Carninci, et al., 2006 ), which are likely determinants in this set. We chose 4 TFs whose binding preferences are not similar: IRF, PU.1, SOX-9 and c-Ets-2 (from TRANSFAC (Matys, et al., 2003)). We also included NF-kappaB (JASPAR(Bryne, et al., 2007) ID MA0061), as this factor is expected to play a major role in immunological response (Bonizzi and Karin, 2004). The positive model for the classifier was designed automatically by evolving e-HMMs with the positive training set. To run the genetic algorithm, we generated 30 HMMs to construct a population where each HMM was composed of 40 blocks including zero blocks and PSSM blocks. The negative model was derived from the positive model by taking PSSM blocks out and performing parameter training with the negative set, as described previously. Figure 6 shows a ROC curve plotting sensitivity (=TP/(TP+FN)) against 1-specificity (=TN/(FP+TN)) of the three-fold cross-validation of the three e-HMM classifiers. To test the evolved model, we selected HMMs after some evolutionary iterations. The three HMMs are acquired from the 80th, 90th and 100th iteration of the simulation. We also tested ensemble of these three HMMs. For comparison, we also applied Cluster-Buster and COMET to identify motif clusters given the same PSSMs. Also, we compared our method with the performance of a classifier using SVMs (Support Vector Machines) (Vapnik, 1995). Using these results, we checked if they can classify the macrophage dataset well by counting the number of predicted sites as in the test by (Chowdhary, et al., 2006). For COMET and Cluster-Buster we regarded a sequence as an induced macrophage promoter if any cluster prediction was made on the sequence. Changes in the cut-off values were used to derive the ROC plots as above. Cluster-Buster performed better than COMET in this test, but had lower sensitivity than the e-HMM classifiers for the whole range of specificity values. The SVM classifier in this test had the worst performance.

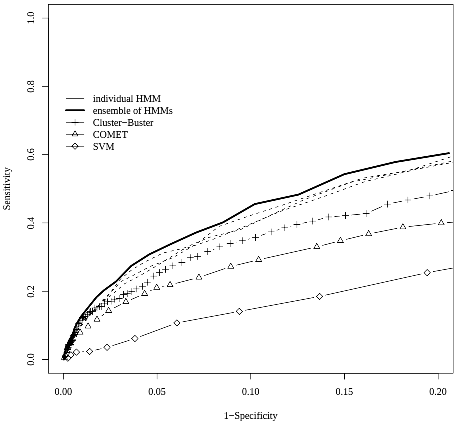

Fig. 6. Performance of e-HMMs as classifiers of macrophage promoters. ROC curves plotting sensitivity versus 1-specificity for three evolved HMMs, ensemble of HMMs, COMET, Cluster-Buster and SVM.

To investigate which grammar the HMMs have found, we decoded the sequence with the evolved HMMs. Table S5 lists the grammar the 3 HMMs found.

## 4 DISCUSSION

In this study, we present a method that models co-occurring motifs in a set of promoters. A wealth of methods taking a set of PSSMs to classify gene sets have been published (Frith, et al., 2001; Frith, et al., 2003; Frith, et al., 2002; Gupta and Liu, 2005; Rajewsky, et al., 2002; Rebeiz, et al., 2002; Sharan and Myers, 2005; Sinha, et al., 2003; Wasserman and Fickett, 1998). The methods range from relatively simple statistical frameworks such as logistic regression (Wasserman and Fickett, 1998) to general classifiers based on Support Vector Machines (Sharan and Myers, 2005) or other more complex methods. However, none of these methods are aimed at describing the inter-dependencies of sites. In our approach, using blocks derived from known transcription factor binding sites as well as blocks modeling the intermediate regions, we build HMM architectures describing the promoters using an evolving model. Thus, we can model the "grammar" of the regulation, i.e., the order of transcription factors, the distance between them, the composition of intermediate regions, etc. This has two immediate advantages. Firstly, it gives increased predictive power for classifying sequences. Secondly, it represents a step towards more realistic descriptions of promoter architecture and function, which is necessary for understanding the complex transcription process. An important issue is that the grammar obtained by e-HMMs may not be the optimal or the simplest one because e-HMMs search for the rules in the sequence set heuristically. Thus, it represents a hypothesis that to some degree can explain the data, and these types of hypothesis can give a starting point for experimental trials. A *cis*-regulatory structure learning algorithm using HMMs was presented by (Noto and Craven, 2007), in which they tried to build logical relationships among binding sites. Starting from a structure with a single motif, they expanded their logics. Our structure learning method is different in that we search for a whole logic using a genetic algorithm. A greedy approach to infer the *cis*-regulatory logic of transcriptional network was suggested using DNA sequence and mRNA expression data (Beer and Tavazoie, 2004). They used a Bayesian network that models combinatorial regulatory rules to predict gene expression pattern of *Saccharomyces cerevisiae*. While the model in this work share some similarity with ours, the study is different as they tried to learn gene expression patterns, which we do not (so, the gene expression data is part of the training data). However, this points to a possible extension of the e-HMM concept – to incorporate expression data in the actual model. Recent studies have claimed that regulatory grammars are very flexible (Brown, et al., 2007) and a predictor can perform well without considering any grammar(Segal, et al., 2008). It is not clear whether this is a universal truth in mammalian gene regulation; the increased classification power of our method compared to others imply that part of grammars are detectable and increase accuracy if taken into account.

A further enhancement introduced is the usage of ensembles of models in classification as well as in motif identification. The ensemble approach is beneficial in that it compensates for a biased result of a single HMM and thus prevents over-fitting the data. As seen in the simulation with the muscle set, the ensemble approach accumulates the results of each genetic step, and generally produces better results than other methods. The concept of e-HMMs can be extended in many directions. For instance, it may be applied to protein sequences. As this methodology is new, many aspects await further study: for instance, the selection of the fitness function, which has implications on search space and the evolution pace within the process. Nevertheless, we believe that methods trying to model the underlying regulatory grammar are an important step at the road from phenomena-based promoter analysis to hypothesis-based.

## FUNDING

This work was supported by a grant from the Novo Nordisk Foundation.

## REFERENCES

Akaike, H. (1973) Information Theory and Extension of the Maximum Likelihood Principle. *In Proc. of the 2nd Int. Symp. on Information Theory*. 267-281.

Beer, M.A. and Tavazoie, S. (2004) Predicting gene expression from sequence, *Cell*, **117**, 185-198.

Berman, B.P., Nibu, Y., Pfeiffer, B.D., Tomancak, P., Celniker, S.E., Levine, M., Rubin, G.M. and Eisen, M.B. (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome, *Proc Natl Acad Sci U S A*, **99**, 757-762.

Bonizzi, G. and Karin, M. (2004) The two NF-kappaB activation pathways and their role in innate and adaptive immunity, *Trends Immunol*, **25**, 280-288.

Brown, C.D., Johnson, D.S. and Sidow, A. (2007) Functional architecture and evolution of transcriptional elements that drive gene coexpression, *Science (New York, N.Y*, **317**, 1557-1560.

Bryne, J.C., Valen, E., Tang, M.H., Marstrand, T., Winther, O., da Piedade, I., Krogh, A., Lenhard, B. and Sandelin, A. (2007) JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update, *Nucleic Acids Res*.

Carninci, P., Sandelin, A., Lenhard, B., Katayama, S., Shimokawa, K., Ponjavic, J., Semple, C.A., Taylor, M.S., Engstrom, P.G., Frith, M.C., Forrest, A.R., Alkema, W.B., Tan, S.L., Plessy, C., Kodzius, R., Ravasi, T., Kasukawa, T., Fukuda, S., Kanamori-Katayama, M., Kitazume, Y., Kawaji, H., Kai, C., Nakamura, M., Konno, H., Nakano, K., Mottagui-Tabar, S., Arner, P., Chesi, A., Gustincich, S., Persichetti, F., Suzuki, H., Grimmond, S.M., Wells, C.A., Orlando, V., Wahlestedt, C., Liu, E.T., Harbers, M., Kawai, J., Bajic, V.B., Hume, D.A. and Hayashizaki, Y. (2006) Genome-wide analysis of mammalian promoter architecture and evolution, *Nat Genet*, **38**, 626-635.

Chowdhary, R., Tan, S.L., Ali, R.A., Boerlage, B., Wong, L. and Bajic, V.B. (2006) Dragon Promoter Mapper (DPM): a Bayesian framework for modelling promoter structures, *Bioinformatics (Oxford, England)*, **22**, 2310-2312.

Crowley, E.M., Roeder, K. and Bina, M. (1997) A statistical model for locating regulatory regions in genomic DNA, *J Mol Biol*, **268**, 8-14.

Durbin, R., Eddy, S., Krogh, H. and Mitchison, G. (1999) *Biological sequence analysis*. Cambridge University Press, Cambridge.

Frith, M.C., Hansen, U. and Weng, Z. (2001) Detection of cis-element clusters in higher eukaryotic DNA, *Bioinformatics (Oxford, England)*, **17**, 878-889.

Frith, M.C., Li, M.C. and Weng, Z. (2003) Cluster-Buster: Finding dense clusters of motifs in DNA sequences, *Nucleic Acids Res*, **31**, 3666-3668.

Frith, M.C., Spouge, J.L., Hansen, U. and Weng, Z. (2002) Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences, *Nucleic Acids Res*, **30**, 3214-3224.

Gupta, M. and Liu, J.S. (2005) De novo cis-regulatory module elicitation for eukaryotic genomes, *Proc Natl Acad Sci U S A*, **102**, 7079-7084.

Krogh, A. (2000) Using database matches with for HMMGene for automated gene detection in Drosophila, *Genome Res*, **10**, 523-528.

Markstein, M., Markstein, P., Markstein, V. and Levine, M.S. (2002) Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo, *Proc Natl Acad Sci U S A*, **99**, 763-768.

Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A.E., Kel-Margoulis, O.V., Kloos, D.U., Land, S., Lewicki-Potapov, B., Michael, H., Munch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S. and Wingender, E. (2003) TRANSFAC: transcriptional regulation, from patterns to profiles, *Nucleic Acids Res*, **31**, 374-378.

Nilsson, R., Bajic, V.B., Suzuki, H., di Bernardo, D., Bjorkegren, J., Katayama, S., Reid, J.F., Sweet, M.J., Gariboldi, M., Carninci, P., Hayashizaki, Y., Hume, D.A., Tegner, J. and Ravasi, T. (2006) Transcriptional network dynamics in macrophage activation, *Genomics*, **88**, 133-142.

Noto, K. and Craven, M. (2007) Learning probabilistic models of cis-regulatory modules that represent logical and spatial aspects, *Bioinformatics (Oxford, England)*, **23**, e156-162.

Rajewsky, N., Vergassola, M., Gaul, U. and Siggia, E.D. (2002) Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo, *BMC Bioinformatics*, **3**, 30.

Rebeiz, M., Reeves, N.L. and Posakony, J.W. (2002) SCORE: a computational approach to the identification of cis-regulatory modules and target genes in whole-genome sequence data. Site clustering over random expectation, *Proc Natl Acad Sci U S A*, **99**, 9888-9893.

Riis, S.K. and Krogh, A. (1996) Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments, *J Comput Biol*, **3**, 163-183.

Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U. and Gaul, U. (2008) Predicting expression patterns from regulatory sequence in Drosophila segmentation, *Nature*, **451**, 535-540.

Sharan, R. and Myers, E.W. (2005) A motif-based framework for recognizing sequence families, *Bioinformatics (Oxford, England)*, **21 Suppl 1**, i387-393.

Sinha, S., van Nimwegen, E. and Siggia, E.D. (2003) A probabilistic method to detect regulatory modules, *Bioinformatics (Oxford, England)*, **19 Suppl 1**, i292-301.

Smale, S.T. and Kadonaga, J.T. (2003) The RNA polymerase II core promoter, *Annu Rev Biochem*, **72**, 449-479.

Stormo, G.D. (2000) DNA binding sites: representation and discovery, *Bioinformatics (Oxford, England)*, **16**, 16-23.

Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., Patapoutian, A., Hampton, G.M., Schultz, P.G. and Hogenesch, J.B. (2002) Large-scale analysis of the human and mouse transcriptomes, *Proc Natl Acad Sci U S A*, **99**, 4465-4470.

Vapnik, V. (1995) *The nature of Statistical Learning Theory*. Springer, New York.

Wasserman, W.W. and Fickett, J.W. (1998) Identification of regulatory regions which confer muscle-specific gene expression, *J Mol Biol*, **278**, 167-181.

Wasserman, W.W., Palumbo, M., Thompson, W., Fickett, J.W. and Lawrence, C.E. (2000) Human-mouse genome comparisons to locate regulatory sites, *Nat Genet*, **26**, 225-228.

Wasserman, W.W. and Sandelin, A. (2004) Applied bioinformatics for the identification of regulatory elements, *Nat Rev Genet*, **5**, 276-287.

Won, K., Prugel-Bennet, A. and Krogh, A. (2006) Evolving the structure of Hidden Markov Models, *IEEE transactions on Evolutionary Computation*, **10**, 39-49.

Won, K.J., Hamelryck, T., Prugel-Bennett, A. and Krogh, A. (2007) An evolving method for learning HMM Structure: prediction of protein secondary structure, *BMC Bioinformatics*, **8**, 357.