

## Sequence analysis

## Identification of transcription factor binding sites with variable-order Bayesian networks

I. Ben-Gal<sup>1,\*</sup>, A. Shani<sup>1</sup>, A. Gohr<sup>2,3</sup>, J. Grau<sup>2,3</sup>, S. Arviv<sup>1</sup>, A. Shmilovici<sup>4</sup>, S. Posch<sup>3</sup>, and I. Grosse<sup>2</sup><sup>1</sup>Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv, 69978, Israel, <sup>2</sup>Institute of Plant Genetics and Crop Plant Research, (IPK), 06466 Gatersleben, Germany, <sup>3</sup>Institute of Computer Science, University Halle, 06099 Halle (Saale), Germany and <sup>4</sup>Department of Information Systems Engineering, Ben-Gurion University, Beer-Sheva, 84105, Israel

Received on September 22, 2003; revised on March 3, 2005; accepted on March 23, 2005

Advance Access publication March 29, 2005

## ABSTRACT

**Motivation:** We propose a new class of variable-order Bayesian network (VOBN) models for the identification of transcription factor binding sites (TFBSs). The proposed models generalize the widely used position weight matrix (PWM) models, Markov models and Bayesian network models. In contrast to these models, where for each position a fixed subset of the remaining positions is used to model dependencies, in VOBN models, these subsets may vary based on the specific nucleotides observed, which are called the context. This flexibility turns out to be of advantage for the classification and analysis of TFBSs, as statistical dependencies between nucleotides in different TFBS positions (not necessarily adjacent) may be taken into account efficiently—in a position-specific and context-specific manner.

**Results:** We apply the VOBN model to a set of 238 experimentally verified sigma-70 binding sites in *Escherichia coli*. We find that the VOBN model can distinguish these 238 sites from a set of 472 intergenic ‘non-promoter’ sequences with a higher accuracy than fixed-order Markov models or Bayesian trees. We use a replicated stratified-holdout experiment having a fixed true-negative rate of 99.9%. We find that for a foreground inhomogeneous VOBN model of order 1 and a background homogeneous variable-order Markov (VOM) model of order 5, the obtained mean true-positive (TP) rate is 47.56%. In comparison, the best TP rate for the conventional models is 44.39%, obtained from a foreground PWM model and a background 2nd-order Markov model. As the standard deviation of the estimated TP rate is  $\approx 0.01\%$ , this improvement is highly significant.

**Availability:** All datasets are available upon request from the authors. A web server for utilizing the VOBN and VOM models is available at <http://www.eng.tau.ac.il/~bengal/>

**Contact:** bengal@eng.tau.ac.il

## 1 INTRODUCTION

One problem in the analysis of DNA sequences is the identification of transcription factor binding sites (TFBSs). The importance of this problem stems from the fact that the combinatorial presence and absence of TFBSs is—to a large degree—responsible for the complexity of gene regulation in virtually every living organism (Wingender *et al.*, 2000, 2001; Pickert *et al.*, 1998; Kel-Margoulis

*et al.*, 2003). The interest in the analysis of TFBSs dramatically grown with the arrival of microarray gene expression data which heralds an important advance in the identification of co-expressed genes (Fickett and Hatzigeorgiou, 1997; Chu *et al.*, 1998; Spellman *et al.*, 1998; Thijs *et al.*, 2001; Ohler and Niemann, 2001; Hanisch *et al.*, 2002). While experimental techniques such as footprinting experiments or chromatin immunoprecipitation experiments allow the identification of TFBSs, these experiments are expensive and time consuming, and require the availability of a well-equipped lab together with professionally trained personnel. Today, the availability of computer hardware installed with cheap or often free bioinformatics software has enabled bench scientists in almost every research lab to run programs such as Match<sup>TM</sup> (Kel *et al.*, 2003) or HMMgene and HMMPro (e.g. Baldi and Brunak, 2001) to predict the location of TFBSs. While bioinformatics identifications are probabilistic in nature and cannot achieve the accuracy of ‘wet’ experimental data, their value lies in the low cost and high speed with which these identifications can be obtained. To expand on the above-mentioned example, Match<sup>TM</sup>, a publicly available and free-of-charge software, can scan several megabytes of DNA for the presence of putative TFBSs within minutes. Hence, TFBS identification programs are extremely popular, despite their limited accuracy.

Most TFBS classification algorithms compute some numerical score reflecting the degree to which a given sequence site matches a given motif. In many TFBS classification algorithms, the underlying scoring model is either a fixed-order Markov model or simply a position weight matrix (PWM) model with no context dependencies at all (e.g. Match<sup>TM</sup>). The latter can be regarded as a fixed-order Markov model of order 0. In the following, we note the main differences among the proposed variable-order Bayesian network (VOBN) model, the PWM model, the fixed-order Markov model and the Bayesian network (BN) model.

Presumably, the most common context-independent model is the PWM (or the position specific score matrix—PSSM). The PWM model has been successfully applied not only to the problem of TFBS classification but also to other diverse problems in DNA and protein sequence analysis (e.g. Salzberg, 1997). Although other scoring models have been able to improve the accuracy of the PWM model in certain cases, they are not as prevalent as the simple PWM model in the classification of TFBSs (Fickett and Hatzigeorgiou, 1997).

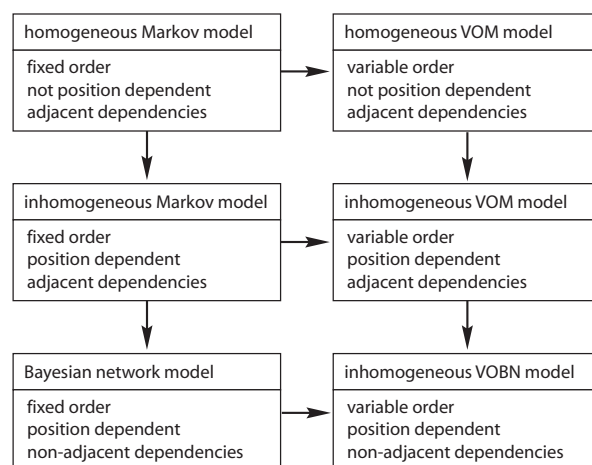
\*To whom correspondence should be addressed.

Hence, many TFBS classification algorithms that are developed today still rely on the PWM model that obtain a relatively good performance, as will be seen below.

The basic assumption of the PWM model is that the basepairs at different positions are statistically independent; hence, the joint probability of finding a multiple-position site factorizes into the product of single-position probabilities (e.g. Djordjevic *et al.*, 2003; Ewens and Grant, 2001; Stormo and Fields, 1998). As indicated in Barash *et al.* (2003) it is an open question whether the ‘strong’ independence assumption of the PWM model is reasonable in view of recent results that point to the dependence between positions (e.g. Benos *et al.*, 2001; Bulyk *et al.*, 2002). This dependence is used by fixed-order models, such as Markov models, hidden Markov models (HMMs) and interpolated Markov models to detect motifs in upstream regions of co-regulated genes (Liu *et al.*, 1995; Ohler *et al.*, 1999; Hughes *et al.*, 2000; Thijs *et al.*, 2001; Ohler and Niemann, 2001; Liu *et al.*, 2001; Salzberg *et al.*, 1998, 1999). Although it is well known (and intuitively clear) that the PWM assumption is violated in almost every TFBS studied to date, this violation, nevertheless, does not prevent the PWM model from being a leading model in the classification of putative TFBSs. In fact, the PWM model, which is based on the (unsupported) independence assumption, is often found to outperform fixed-order Markov models of higher order that are based on the (reasonable and supported) dependence assumption.

In this paper, we show that the above-mentioned contradiction is due to the unbalanced comparison between the PWM model and a high-order Markov model with respect to their number of parameters. Although the PWM independence assumption is unsound and results in an under-fitted model with a smaller-than-necessary number of parameters, it often outperforms fixed-order Markov models that tend to be over-fitted due to their large dimensionality, given the limited amount of training data. The solution presented here is based on the development of a variable-order model that, in terms of its order, stands in between these two types of models. We show that the variable-order models do not ignore the statistical dependencies between basepairs, yet, they take into account only those dependencies that are found to be statistically significant.

As an extension to fixed-order models, we suggest using the inhomogeneous VOB model. The VOB model is a generalization of the variable-order Markov (VOM) model, which was originally proposed for data compression (Rissanen, 1983) and later applied in various forms to prediction and identification (Weinberger *et al.*, 1995), statistical process control for finite-state processes (Ben-Gal *et al.*, 2003), text clustering (Vert, 2001), modelling of genetic texts, including TFBSs and protein coding regions (Ron *et al.*, 1996; Buhlmann and Wyner, 1999; Orlov and Potapov, 2000; Orlov *et al.*, 2002; Bilu *et al.*, 2002, <http://www.cs.huji.ac.il/~johnblue/papers/>; Zhao *et al.*, 2004) and modelling of protein families (Bejerano and Yona, 2001). In contrast to fixed-order Markov models, where the order is the same for all positions and for all contexts, in VOM and VOB models, the order may vary for each position based on its context. There are three main differences between the above VOM models and the proposed VOB model. The first are variations in the construction of the models as seen in section 2. The second is the use of *inhomogeneous* (position-dependent) VOB models versus the homogeneous VOM model—a crucial property for the classification of *Escherichia coli* TFBSs. The third is the VOB generalization to contexts from non-adjacent positions in a manner similar to BN models, which are discussed next.



**Fig. 1.** The generalization hierarchy of the variable-order models. The generalized variable-order Bayesian network (VOBN) model can be interpreted as a generalization of the VOM model as well as of the Bayesian network (BN) model.

The BN model is a graphical representation of probabilistic dependence knowledge (Pearl, 1988) that was applied to the analysis of gene expression data (Friedman *et al.*, 2000), genetic linkage analysis (Heckerman *et al.*, 1995) and the identification of TFBSs and other functional DNA regions (Cai *et al.*, 2000; Barash *et al.*, 2003; Castelo and Guigo, 2004). In the BN graph an edge is directed from an influencing position (parent) into an influenced position (child). In contrast to fixed-order Markov models, in BN models it is not assumed that dependencies are necessarily between adjacent positions. The difference between BN models and the proposed VOB is that, in general, the order of the model in BNs depends only on the size of the parent's subset, while in VOB models it also depends on the specific nucleotides observed in each parent subset. As a result, the number of parameters that need to be estimated from the data is substantially smaller, yielding a smaller chance for over-fitting of the model to the training dataset. A class of models which are closely related to VOB models is the context-specific Bayesian networks (CSBNs) (e.g. Boutilier *et al.*, 1996; Friedman and Goldszmidt, 1996). The main differences between CSBNs and the proposed VOB models are in the method of encoding and constructing the context-specific dependencies (e.g. complete versus non-complete trees), parameter estimation and refinement methods, and the manner in which the context-dependencies are integrated in the model-learning phase (e.g. starting with an over-fitted model), as described below.

The proposed VOB model is a true generalization (rather than a replacement) of PWM, fixed-order Markov and BN models in the sense that these models are special cases of the VOB model. This means that in cases where the statistical dependencies are insignificant, the VOB model ‘automatically’ degenerates to the PWM model. If dependencies exist only amongst adjacent positions and the ideal memory length for a position is identical for all basepairs, the VOB generalizes to a fixed-order Markov model. Similarly, if the ideal memory length for a given position is identical for all basepairs and depends only on the number of parents, the VOB generalizes to a BN model (Fig. 1).

In the remainder of the paper we study the degree to which the VOBN yields useful generalizations of the PWM and the fixed-order Markov models in the context of TFBS classification.

## 2 METHODS

In this section we introduce the homogeneous and inhomogeneous models, which we will apply, respectively, for modelling the DNA background not containing TFBSs and for the TFBSs. We start with a homogeneous Markov model of order zero, continue with homogeneous fixed-order Markov models and end with homogeneous VOM models as generalizations of fixed-order Markov models. We then introduce the VOM tree and outline the VOM construction algorithm. Subsequently, we discuss different inhomogeneous models, starting with the PWM and inhomogeneous VOM models and ending with the proposed VOBN models. Finally, we introduce the classification rule used, the stratified-holdout procedure used and the performance measure by which we quantify the classification accuracy.

### 2.1 Homogeneous models

In the following, we adopt some definitions and notations from Buhlmann and Wyner (1999), Ohler *et al.* (1999), and Ben-Gal *et al.* (2003). Let  $x_l^n = x_l, x_{l+1}, \dots, x_{n-1}, x_n$  define a sequence with  $n - l + 1$  symbols over a finite alphabet  $X$  of cardinality  $|X| = d$ . In case of the TFBS classification problem,  $d = 4$ ,  $X = \{A, C, G, T\}$  and  $x_j$  is the nucleotide at position  $j$ , with  $1 \leq j \leq N$  in the DNA sequence  $x_1^N$  of length  $N$ . The likelihood of sequence  $x_1^N$  is computed by the multiplication chain rule (e.g. Ohler *et al.*, 1999):

$$P(x_1^N) = \prod_{j=1}^N P(X_j = x_j | X_1^{j-1} = x_1^{j-1}), \quad (1)$$

where  $P(\cdot)$  stands for probability,  $X_j$  is the random variable representing the nucleotide at position  $j$  with  $x_j$  as its realization,  $X_1^{j-1}$  is a sequence of random variables for the context with  $x_1^{j-1}$  as the actually observed context of the symbol  $x_j$  containing its preceding  $j - 1$  symbols and  $x_1^0$  is the empty string.

Traditional models often consider only a small part of the available context (or no context) in order to minimize the number of parameters to be estimated and to avoid over-fitting the training dataset. Many papers suggest a homogeneous zeroth-order Markov model (called also a Bernoulli model) as a simple background model (e.g. Liu *et al.*, 1995; Hughes *et al.*, 2000). The likelihood for this model, which we abbreviate by Markov(0), is computed by multiplying the probabilities of the symbols, i.e.

$$P(x_1^N) = \prod_{j=1}^N P(X_j = x_j), \quad (2)$$

because this model considers no context at all. Using a zeroth-order homogeneous model implies that  $P(\cdot)$  is identical for all  $j$ . Other studies indicate that such a model poorly reflects the complex structure of genome sequences (Thijs *et al.*, 2001; Liu *et al.*, 2001) and suggest higher-order models. Accordingly, in  $L$ th-order Markov models, denoted here by Markov( $L$ ), the likelihood of the sequence depends on the sequence of predecessors of a fixed length  $L < N$ , i.e.,

$$P(x_1^N) = \prod_{j=1}^N P(X_j = x_j | X_{j-L}^{j-1} = x_{j-L}^{j-1}), \quad (3)$$

where  $x_{j-L}^{j-1} = x_1^{j-1}$  if  $j - L \leq 0$ ; i.e., the memory length cannot exceed the number of the preceding symbols and the subscript  $j - L \equiv \max(j - L, 1)$ .

As indicated in Buhlmann and Wyner (1999) and Orlov *et al.* (2002), a major problem of fixed-order Markov models is that the number of model parameters grows exponentially with the model order  $L$ , resulting in a very sharp and discontinuous transition from under-fitted models (that do not capture enough statistical dependencies in the data) to over-fitted models (that contain too many parameters). For example, the number of free parameters in fixed-order Markov models with  $d = 4$  and  $L = 2, 3, 4$ , and 5 is equal to 63, 255, 1023 and 4095 respectively.

Some approaches to solving the above-mentioned problems look for the optimal  $L$ , which maximizes the likelihood of the training dataset, or apply the interpolation of different model orders, as suggested in Salzberg *et al.* (1998, 1999) and Ohler *et al.* (1999). The difficulty with these approaches is that the model order is averaged or weighted over different subsequences in the training set, and thus might be either too short or too long for different symbols in the set. Symbols along the sequence might depend on contexts that are shorter than an averaged  $L$ , even if it has a relatively small value. We suggest allowing a variable model order  $L_j$  that depends on the preceding symbols to position  $j$ ; thus, the order of the Markov model becomes a function of the context at each position,

$$P(x_1^N) = \prod_{j=1}^N P(X_j = x_j | X_{j-L_j}^{j-1} = x_{j-L_j}^{j-1}), \quad (4)$$

where the variable-order  $L_j = L(x_{j-1}, x_{j-2}, \dots)$  is itself a function of the preceding symbols. An optimal value for  $L_j$  defines the shortest context for which the transition probability of symbol  $x_j$  is practically equal to the transition probability of that symbol given the context of maximal order  $L$ , i.e.,

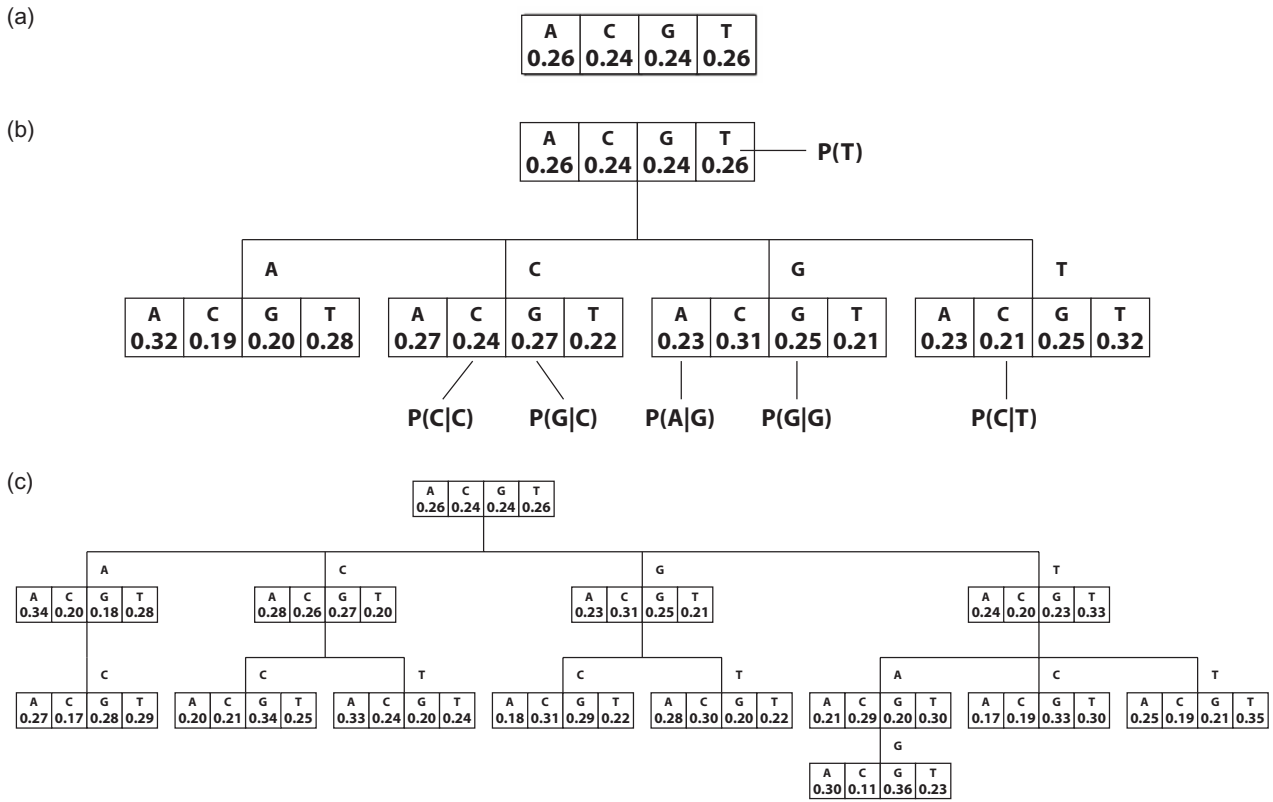
$$L_j = \min \left\{ \tilde{L} \mid P(X_j = x_j | X_{j-\tilde{L}}^{j-1} = x_{j-\tilde{L}}^{j-1}) = P(X_j = x_j | X_{j-L}^{j-1} = x_{j-L}^{j-1}) \right\}. \quad (5)$$

Note from Equation (3) that for the fixed-order Markov chain  $L(x_{j-1}, x_{j-2}, \dots) = L$  for all  $x_j$ , whereas, for the suggested variable-order Markov model,  $L_j \leq L$ , implying that some transition probabilities of the Markov chains can be lumped together (e.g. Buhlmann and Wyner, 1999; Orlov and Potapov, 2000).

### 2.2 The context tree representation

Variable-order Markov models, including fixed-order Markov models, can be represented by a tree, which Rissanen (1983) calls *context tree* (called also VOM tree in this paper).

For illustration purposes, let us start with simple examples of homogeneous VOM trees that will then be defined more formally. The trees were constructed from the intergenic background dataset described in Section 3. Figure 2a represents a (degenerated) tree that consists of a single root that is equivalent to a Markov(0) model. The root contains four probabilities for nucleotides  $A, C, G$ , and  $T$ , respectively. In this case the model has no memory, and the likelihood is computed by multiplying the probabilities of the nucleotides, as indicated in Equation (2). Figure 2b represents a more developed tree that consists of a single root with four leaves. Each node in the tree contains four parameters—the conditional probabilities of nucleotides—ordered as  $P(A|x_{j-1}), P(C|x_{j-1}), P(G|x_{j-1}), P(T|x_{j-1})$ , where  $x_{j-1}$  is the context corresponding to each of the nodes. The tree is equivalent to a first-order Markov model. The tree root contains the (unconditional) nucleotide probabilities, while the leaves contain 16 conditional probabilities for each nucleotide given a preceding nucleotide. The likelihood is computed by multiplying the transition probabilities of the symbols given the previous symbol, as shown in Equation (3) with  $L = 1$ . For example,  $P(TCCGGA) = P(T) \times P(C|T) \times P(C|C) \times P(G|C) \times P(G|G) \times P(A|G) = 0.26 \times 0.21 \times 0.24 \times 0.27 \times 0.25 \times 0.23$ . In a similar manner, a tree that represents a fixed-order Markov( $L$ ) model contains  $d^L$  leaves. Figure 2c represents a VOM tree that consists of 14 nodes. In this case, the depth of the tree is no longer fixed. All branches of the original tree (not presented), which corresponds to a 5th-order Markov model, were pruned by the VOM construction algorithm (see below). Note that the tree branches from the root on top down to the leaves represent the *reversed* contexts. Thus, an extension of a branch by adding a node represents the extension of a context by an *earlier* observed symbol (Buhlmann and Wyner, 1999; Ben-Gal *et al.*, 2003). For example, the first level node for context  $A$  indicates a 1st-order Markov model, which is used instead of the *longer contexts* for all nucleotides except for context  $C$ . The single node at level 3 represents a 3rd-order Markov model that consists of the context  $GAT$ . Although the



**Fig. 2.** (a) A degenerated VOM tree represented by a root node, which is equivalent to a Markov(0) model. The tree is constructed from the intergenic background dataset. The label of the node is the 4-dimensional probability vector of the single-nucleotide probabilities  $P(A)$ ,  $P(C)$ ,  $P(G)$  and  $P(T)$  in this order. Note that in this and the following figures the sum of probabilities in a node might not sum exactly to 1 due to round off errors. (b) A homogeneous unpruned VOM tree, which is equivalent to a Markov(1) model. The tree is constructed from the intergenic background dataset. The leaf nodes are labeled with the transition probability vector given a single-nucleotide context. The root node is labeled with the unconditional probability vector of nucleotides. (c) The homogeneous VOM(5, 0.65) tree. The tree is constructed from the intergenic background dataset. The nodes are labeled with the transition probability vector given the context defined by the reverse path from the nodes to the root. The root node is labeled with the unconditional probability vector of nucleotides.

maximal order allowed in this case is 5, all branches are pruned to a lower order. As a result, instead of using a full Markov model of order  $L = 5$  with  $d^{L+1} - 1 = 4095$  free parameters, the VOM model in this example has only  $14 \times 3 = 42$  free parameters. The likelihood of a sequence given a VOM model depends on the contexts of a varying-order  $L_j$ , as seen in Equation (4). Using the above example,  $P(TCCGGA) = P(T) \times P(C|T) \times P(C|TC) \times P(G|CC) \times P(G|CG) \times P(A|G) = 0.26 \times 0.20 \times 0.24 \times 0.34 \times 0.29 \times 0.23$  (the difference in the probabilities in level 1 compared to Fig. 2b stems from the probability adjustment explained in the pseudo code below). Thus, from the fourth up to the sixth nucleotide, the order of the model, as represented by the equivalent branches in the tree, is smaller than the number of preceding symbols. For example, the tree does not contain the full-order branch for  $TCCGG$ , since for the last nucleotide  $P(A|TCCGG) \approx P(A|G)$ . Let us now define the VOM tree more formally.

The VOM model assigns a context for each element in the sequence, and defines the transition probability of each symbol  $x_j$  given its context. Graphically, the VOM tree has a root node on top, from which the branches are developed downwards, with the constraint that each internal node has at most  $d$  children, with differently labelled edges. The tree is not necessarily balanced (i.e. not all the branches need to be of the same length) or complete (i.e. not all the nodes need to have  $d$  children). Each node contains  $d$  transition probabilities of symbols given the context, which is represented by the reversed path from that node to the root (this is why these trees are also called *suffix trees*). *Optimal contexts* that satisfy Equation (5) are represented by the reversed path  $x_{j-L_j}, x_{j-L_j+1}, \dots, x_{j-1}$  from the leaves to the root.

Sometimes an optimal context is represented by a *partial leaf*, which is an *internal* node whose reversed path satisfies Equation (5) for some (but not all) nucleotides (Ben-Gal et al., 2003; Buhlmann and Wyner, 1999).

### 2.3 Construction of the VOM tree

In the following, we describe the algorithm that we use for the construction of the VOM trees for sigma-70 sites (foreground set) and non-sigma-70 sites (background set). The algorithm consists of two main stages. First, it constructs a complete and balanced tree of depth  $L$ , which corresponds to a fixed-order Markov model of order  $L$ . Second, it iteratively prunes the tree by a backward procedure. The initial order  $L$  of the models is estimated using the number of samples in the training sets, such that on average each leaf contains at least 10 data points (in our case approximately 40 data points per leaf). Once an initial (complete and balanced) tree of order  $L$  is constructed, its probability parameters are estimated (as denoted by the tilde sign) by the frequencies of the respective subsequences, i.e.,

$$\tilde{P}(X_j = x_j | X_{j-L}^{j-1} = x_{j-L}^{j-1}, \Omega_k) = \frac{n_k(x_{j-L}^j)}{n_k(x_{j-L}^{j-1})}, \quad (6)$$

where  $n_k(\cdot)$  denotes the frequency of its argument in a training set taken from class  $\Omega_k$ :  $\Omega_1$  is the class of TFBSs (the foreground set) and  $\Omega_2$  is the class of non-TFBSs (the background set). To compensate for zero occurrences of certain oligonucleotides, we use a pseudo-count, which is added to all frequencies. The value of the pseudo-count depends on the level of the node

**Table 1.** PWM and consensus sequences for the ‘−35 box’ and the ‘−10 box’, as derived from the sigma-70 foreground dataset

	1	2	3	4	5	6	1	2	3	4	5	6
A	0.10	0.07	0.10	<b>0.54</b>	0.17	<b>0.49</b>	0.07	<b>0.74</b>	0.15	<b>0.57</b>	<b>0.53</b>	0.08
C	0.10	0.09	0.13	0.19	<b>0.54</b>	0.14	0.12	0.07	0.12	0.12	0.22	0.07
G	0.12	0.09	<b>0.57</b>	0.12	0.12	0.17	0.10	0.07	0.14	0.16	0.09	0.05
T	<b>0.68</b>	<b>0.74</b>	0.20	0.15	0.18	0.20	<b>0.72</b>	0.12	<b>0.58</b>	0.15	0.15	<b>0.80</b>
	T	T	G	A	C	A	T	A	T	A	A	T

in the tree. For the background model we assure that the sum of all pseudo-counts in a level is 4096; for the foreground model the sum is 16. This rule corresponds to an equivalent sample size (ESS) of 4096 and 16, respectively, (i.e. each oligonucleotide obtains the same pseudo-count) as used in Bayesian networks (Heckerman *et al.*, 1995). It results in an effective pseudo-count of 1024 for each frequency in the root and a pseudo-count of 1 in the leaves of a 5th-order tree as a background model.

Then, we compute the Kullback–Leibler (KL) divergence (Kullback, 1959) of the conditional probabilities of symbols between each leaf and its parent node. If the KL divergence is smaller than a pre-selected pruning threshold, the leaf is pruned. A small KL divergence implies that there is no significant divergence in the symbol distribution when using the reduced order of the model, or in other words, that the larger model order, which is represented by the leaf, does not add much information and can be pruned without affecting the likelihood significantly. The pruning procedure is repeated until the KL divergence is greater than the predefined pruning threshold for all the leaves in the tree.

A simple pseudo-code to construct a VOM tree from a given (supervised) training set is now detailed. In our example we train two independent VOM models, a foreground VOM model on TFBS oligonucleotides and a background VOM model on non-TFBS oligonucleotides (for simplicity of presentation, we omit the class notation,  $\Omega_k$ , since the algorithm is applied to classes independently).

- (1) Construct an initial complete and balanced Markov tree of a maximal fixed-order  $L$  such that on average, each leaf counter contains 10 data points. This rule yields  $L = 5$  for the intergenic background set and  $L = 1$  for the foreground set.
- (2) Estimate the conditional probabilities in the tree nodes by using Equation (6) and by adding a pseudo-count  $\varepsilon = ESS/(d \cdot d')$  to all counters, with  $t$  being the depth of the node, starting from the root with  $t = 0$ .
- (3) Estimate the KL divergence of the distribution of nucleotides between leaves and their parent nodes:

$$\Delta_{\text{leaf}} = \sum_{x_j \in X} \tilde{P}(x_j | x_{j-L_j}^{j-1}) \log_2 \left( \frac{\tilde{P}(x_j | x_{j-L_j}^{j-1})}{\tilde{P}(x_j | x_{j-L_j+1}^{j-1})} \right),$$

for all leaves.

- (4) Prune the leaves with a small KL divergence. The pruning process is executed bottom-up from each leaf to the root according to the following rule: If  $\Delta_{\text{leaf}} \leq c \cdot \psi$ , where  $c$  is a predefined pruning parameter, prune that leaf by setting  $L_j = L_j - 1$ . For the foreground model we use  $\psi = 1$  and for the background model we use  $\psi = d^{t+1}/n(x_{j-L_j}^{j-1})$ . For such a  $\psi$ , the deeper a node is in the tree, the easier it is to prune it, and the more samples that reach that node, the harder it is to prune it. Otherwise, set  $L_j$  as the optimal order for that leaf.
- (5) If all leaves are left unpruned, stop. Otherwise, go back to step 3 and repeat for all of the pruned leaves.
- (6) Refine the probability parameters in the obtained VOM tree by introducing the pseudo-counts and subtracting the counts of each symbol

in the descendent nodes from the count of that symbol in the parent node respectively, i.e.  $n_k(x_j | x_{j-L_j+1}^{j-1}) - n_k(x_j | x_{j-L_j}^{j-1})$  for further details, see Ben-Gal *et al.* (2003); Buhlmann and Wyner (1999).

Note again that in general  $L_j$  can be equal for all positions, as in the case of a fixed-order Markov model, including the zeroth-order Markov model with  $L_j = 0$ .

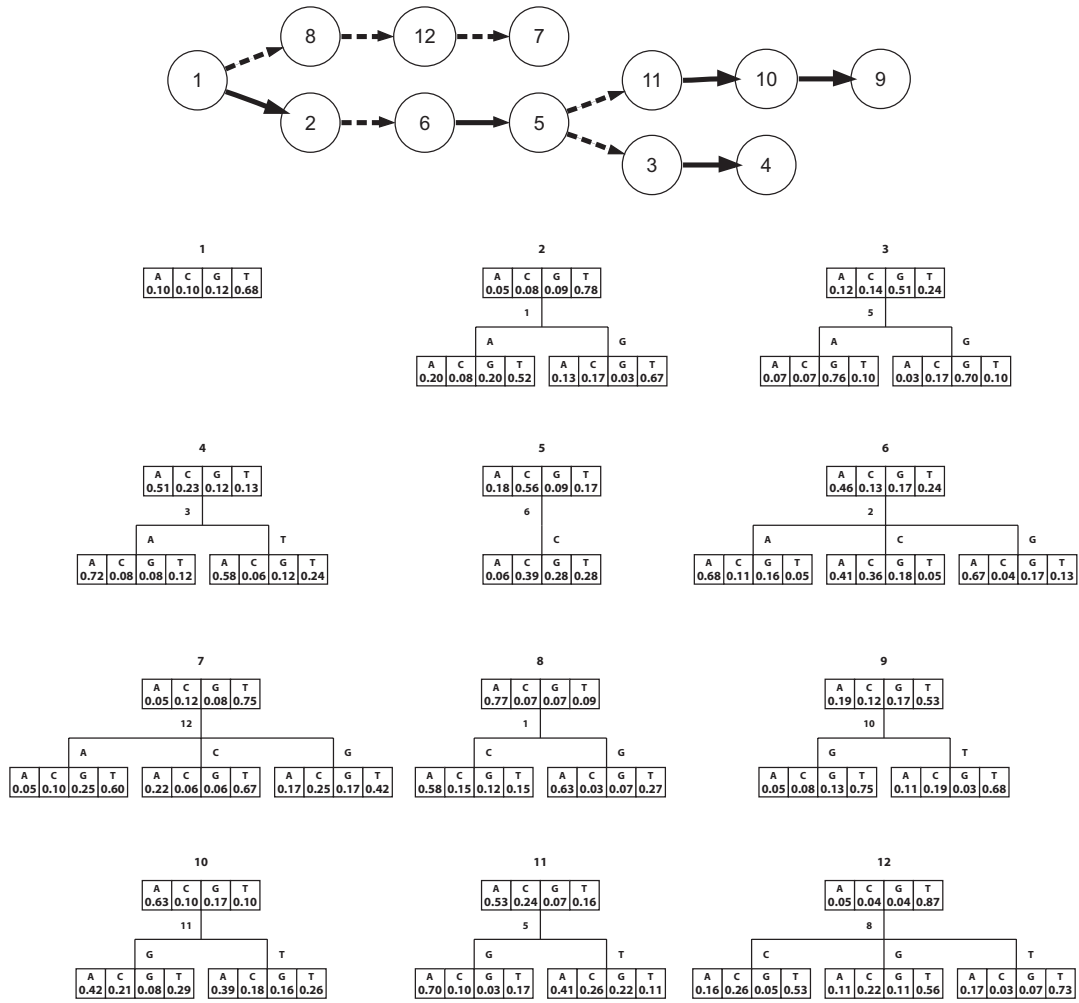
There are several variations for the construction of VOM trees (e.g. Rissanen, 1983; Buhlmann and Wyner, 1999; Orlov *et al.*, 2002; Ben-Gal *et al.*, 2003) that might affect their classification performance significantly. For example, Orlov *et al.* (2002) use homogeneous (rather than inhomogeneous) VOM trees to model (rather than classify) TFBS oligonucleotides. Their initial tree depth is set to 10 and does not depend on the size of the dataset. They use different pseudo-counts and prune the VOM tree based on a stochastic complexity measure, which is related to the KL divergence, which we use, yet penalizes directly the model complexity. Instead, to avoid model over-fitting, we use Equation (5) and search for a good value of the pruning constant over a set of stratified-holdout experiments. Finally, Orlov *et al.* (2002) do not refine the probability parameters as we do in step 6 above.

## 2.4 Inhomogeneous models

The TFBS oligonucleotides are often represented by inhomogeneous models where a model is built for each position in the sequence. The PWM is possibly the most popular inhomogeneous model for binding sites (Fickett and Hatzigeorgiou, 1997; Ewens and Grant, 2001; Mount, 2001; Barash *et al.*, 2003). The underlying independence assumption for this model implies that the likelihood of a sequence can be computed by Equation (2) with a distinction that the marginal probability  $P(X_j = x_j)$  is estimated for each position independently—based on the nucleotide frequencies at that position only. Table 1 presents the probability parameters of the PWM model for the two hexamers found respectively in the ‘−35 box’ and the ‘−10 box’, as estimated from the sigma-70 foreground dataset. The last row in the table presents the consensus sequence. We use this PWM model as a reference model to the proposed inhomogeneous VOM tree, which is described next.

The construction of a foreground inhomogeneous VOM tree is performed by applying the VOM pseudo code outlined above to each position of the set of aligned binding sites. Thus, we construct 12 independent VOM trees, one for each of the 12 positions of the sigma-70 binding site.

The proposed VOB model extends the inhomogeneous VOM model by allowing dependencies between non-adjacent positions. To obtain a VOB model we first learn a BN model, where a node corresponds to a position, and a directed edge (an arrow) from node  $i$  to node  $j$  means that the nucleotide at position  $i$  is considered part of the context of the nucleotide at position  $j$ . Then, for each position the VOM tree is constructed given the context of its parents in the graph. Subsequent to the trees’ construction, we prune and adjust probabilities in every tree as in the original VOM model. Minor modifications are necessary in order to adapt Equation (5) for the VOB model. The preceding symbols to position  $j$  in the sequence  $X_{j-L}^{j-1}$  are replaced by the  $L$  parents in the BN dependence graph, denoted by  $Pa(X_j)_L \equiv (Pa(X_j)_1, Pa(X_j)_2, \dots, Pa(X_j)_L)$ , where for every  $i$  and given a dependence measure  $D(\cdot, \cdot)$ :  $D(Pa(X_j)_i, X_j) \geq D(Pa(X_j)_{i+1}, X_j)$ . We use *mutual information* between the positions as the dependence measure.



**Fig. 3.** The BT/BN(1) dependence graph and the corresponding VOB(1, 2<sup>-3.75</sup>) forest, as constructed from the sigma-70 foreground dataset. In the dependence graph (top) the solid lines represent adjacent positions and the dashed lines represent non-adjacent positions. In the VOB forest, the nodes are labeled with the unconditional probability vector of nucleotides at that position in the box. The position is denoted by the upper numerical label (above the root). The position in the context of which we split is labeled by the numerical label near the split. Nodes 1–6 represents the ‘–35 box’ and nodes 7–12 represents the ‘–10 box’.

The equivalent equation to Equation (5) for optimal order of positions in the VOB model is

$$L_j = \min \left\{ \tilde{L} \left( P \left( X_j = x_j \mid Pa(X_j)_L^1 = pa(x_j)_L^1 \right) \right. \right. \\ \left. \left. = P \left( X_j = x_j \mid Pa(X_j)_L^1 = pa(x_j)_L^1 \right) \right\}. \quad (5a)$$

For the case of Bayesian tree (BT) models, an efficient algorithm exists to learn the maximum likelihood graph structure from the data. This algorithm is equivalent to finding the maximum spanning tree over a fully connected, undirected graph using positions as nodes and mutual information between positions as edge weights (Chow and Liu, 1968). For first-order VOB models we use the dependencies learned with the BT algorithm (e.g., Prim’s algorithm). Figure 3 shows a first-order VOB model constructed from the foreground dataset of 238 sigma-70 binding sites. This example illustrates an important property of the VOB model: since insignificant branches are pruned, the tree can serve as a compact and accurate exploratory tool for the dependencies among the basepairs in the sequence. In Section 4, we comment on some of the dependencies observed in the sigma-70 dataset. Note that pruning a VOB model is not equivalent to pruning edges of a BN dependence graph, as the VOB pruning is context-specific and allow for partial

leaves. The differences in the probabilities of symbols between the forest roots and the PWM model result from the refinement process described in the pseudo-code above. The performance of the VOB model is further studied in Section 4.

## 2.5 Classification rule

Once the foreground and background models are selected, the parameters are estimated and refined from a training set, and the log-likelihood ratio of these models is used for classification. In this TFBS classification, a site is declared as a TFBS if the log-likelihood ratio is greater than a given threshold  $T$ ; i.e. if

$$\log_2 \frac{P(x_1^N | \Omega_1)}{P(x_1^N | \Omega_2)} \geq T. \quad (7)$$

Specific biological knowledge, such as the low number of TFBS sequences in the genome, can be taken into account to specify the threshold value  $T$  to guarantee the balance of the specificity and the sensitivity of the classifier. In the experiments presented in Section 4 we choose a value of  $T$  that guarantees a true-negative (TN) rate of 99.9%. The *E.coli* genome consists of approximately  $4 \times 10^6$  basepairs (bp) containing approximately 4000 genes. As we expect one gene per 1000 bp, a TN rate of 99.9% keeps the number of

false predictions smaller than the expected number of true sigma-70 sites for whole-genome analyses.

## 2.6 The performance measure

Our main performance measure is the mean true-positive (TP) rate (also known as *sensitivity*), which is the ratio of the number of TPs and all positive samples, for replicated stratified-holdout experiments having a fixed TN rate of 99.9%. We also compute the standard deviation of the estimated mean TP rate as a measure for the model robustness and the dependence of its accuracy on the training dataset.

## 2.7 Stratified-holdout sampling

In order to minimize over-fitting effects, we conduct a  $10^6$ -fold stratified-holdout experiment by iteratively applying the following procedure. A random 10% of all TFBS sequences are excluded from the foreground dataset and a model is constructed based on all of the remaining sequences in the dataset. A random 10% of all background sequences are excluded from the background dataset and a model is constructed based on all other remaining sequences in the set. Based on these models the likelihood of every sequence in the excluded foreground and background sequences is calculated. Using the model likelihoods, the score (7) is computed, according to which each excluded foreground sequence is marked as either TP or FN, and each 12-mer of each excluded background sequence is marked as either TN or FP.

## 3 DATASETS

Throughout the experiments we use three datasets: one foreground dataset that contains 238 carefully selected *E.coli* sigma-70 binding sites of length 12 bp, and two background datasets—one that contains randomly permuted TFBS sequences and another that contains 12-mers sampled from 472 intergenic ‘non-promoter’ sequences in *E.coli*.

### 3.1 The sigma-70 foreground dataset

Transcription initiation in *E.coli* is controlled to a large degree by the binding of the RNA polymerase holoenzyme together with multiple cofactors to the promoter region just upstream of the transcription start sites. One of the cofactors, which is believed to convey a large fraction of the DNA binding specificity, is the sigma-70 factor. Two well-conserved *cis* elements, the ‘–35 box’ and the ‘–10 box’, can be found —~35 and ~10 bp upstream—to the transcription start site of many *E.coli* mRNA genes. In order to obtain a dataset of these *cis*-element pairs (and likely sigma-70 factor binding sites) we perform the following steps:

- We start with a dataset of 300 binding site pairs from PromEC (Margalit *et al.*, <http://bioinfo.md.huji.ac.il/marg/promec/index.html>). Each of these binding site pairs consists of two hexamers, so the motif length for all of the models discussed in this paper is  $L = 12$ .
- We remove all binding site pairs that could not be found in the database RegulonDB 3.0 (Salgado *et al.*, 2000) created by Julio Collado-Vides and coworkers, or which are not annotated there as sigma-70 binding sites.
- We map the remaining binding site pairs to the *E.coli* genome, including the ‘spacer’ sequences between the two hexamer boxes, and remove all binding site pairs that could not be mapped uniquely to the *E.coli* genome (Blattner and Schroeder, 1984), or that got mapped to a protein-coding region (according to the NCBI annotation).

Following the above procedure we obtain a dataset of 238 binding site pairs, which we call the ‘sigma-70 foreground dataset’, or simply the

‘foreground dataset’. We choose the above-mentioned very stringent rules to derive the ‘sigma-70 foreground dataset’ for a simple reason: it is generally true that, for statistical analyses, a small dataset of high quality is more valuable than a larger dataset of lower quality. Hence, we would like to obtain a foreground dataset with a minimum amount of contamination, and we are willing to sacrifice TP binding site pairs in order to guarantee a very low number of false-positive binding site pairs in the foreground dataset.

### 3.2 The background datasets

We generate two background datasets—the ‘random background set’ and the ‘intergenic background dataset’—for two different studies. We use the random background dataset in order to study the degree to which the VOM/VOBN models capture the existing statistical dependencies in the sigma-70 foreground dataset. For this study it is important (i) to eliminate possible correlations in the sequences of the background dataset; and (ii) to eliminate a possible classification success simply due to a different nucleotide composition in the foreground and the background datasets. We use the intergenic background dataset in order to study the degree to which higher-order background models can improve the classification of sigma-70 binding site pairs versus 12-mers sampled from intergenic regions.

**3.2.1 The random background dataset** The homogeneous zeroth-order Markov model is a popular background model of intergenic sequences (Liu *et al.*, 1995; Neuwald *et al.*, 1995; Hughes *et al.*, 2000; Thijs *et al.*, 2001). In order to generate a dataset without ‘built-in’ statistical dependencies among different positions, and in order to eliminate any composition difference between the foreground and the background sets, we generate the random background dataset as follows: Learn a zeroth-order Markov model from the foreground dataset. Use this Markov model to generate 427 random sequences of length 182, which gives approximately the same number of 12-mer windows as in the intergenic background dataset.

**3.2.2 The intergenic background dataset** Many experiments have been performed to obtain sigma-70 factor binding sites in the *E.coli* genome, but only little work has been devoted to identify—with experimental rigour—sequences that are free of sigma-70 binding sites. Hence, we adopt the following protocol (Gelfand, personal communication; Thijs *et al.*, 2001) to obtain sequences that are unlikely to contain sigma-70 binding sites. Two neighbouring genes are located either on the same strand or on opposite strands. If they are located on opposite strands, they either overlap or share a common intergenic region. If they share a common intergenic region, that region is either a common 5′ (or upstream) intergenic region of both genes, or a common 3′ (or downstream) region of both genes. Provided that the gene annotation is reliable, the common 3′ (or downstream) intergenic regions between two neighbouring genes should not contain sigma-70 binding sites. We extract the set of common 3′ (or downstream) intergenic regions between two neighbouring genes from the complete *E.coli* genome (according to the NCBI annotation), and obtain a dataset consisting of 472 sequences with a total of 77 644 nucleotides, which we call the intergenic background dataset.

## 4 RESULTS AND DISCUSSION

The analysis of the above-mentioned data is performed in three stages. First, we study the degree to which the VOB model is capable of capturing statistically significant (and perhaps

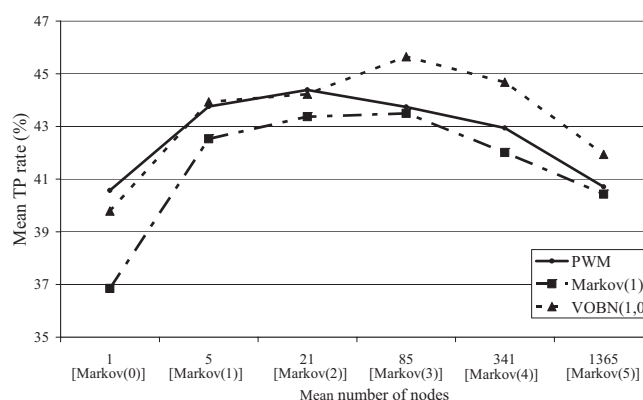


biologically relevant) dependencies among the different positions within sigma-70 factor binding sites, compared to inhomogeneous fixed-order Markov models. Since we focus on the contribution of the foreground set, we use the random background dataset and find the zeroth-order Markov model as the best background model when we train the VOM model on this dataset. Second, we study the degree to which statistical dependencies present in the intergenic background dataset can improve the classification performance; hence, we increase the order of the homogeneous Markov models for the background from  $L = 0$  to  $L = 5$ . Third, we apply VOB models constructed with different pruning constants to the 'foreground dataset' and VOM models constructed with different pruning constants to the intergenic background dataset, and compare the accuracy of these models with the accuracy of fixed-order Markov models including the PWM model and BTs.

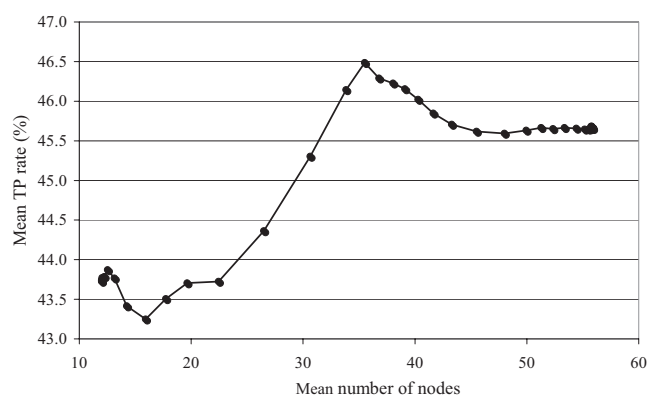
We briefly describe the notations used in the following figures and discussion.

- (1) *Inhomogeneous models*: The model constructed from the sigma-70 foreground dataset. Two types of inhomogeneous models are considered:
  - (a) Markov( $L$ )—the inhomogeneous Markov model of order  $L$ , which includes the PWM model in case of  $L = 0$ .
  - (b) VOB( $L, c$ )—the generalized VOM model with  $L$  denoting the initial maximal order and  $c$  denoting the pruning constant, equivalent to the PWM model in the case of  $L = 0$  and  $c = 0$ . The VOB( $L, c$ ) model is equivalent to a BN model of order  $L$  for  $c = 0$ . We frequently use a BT model—equivalent to a VOB(1,0)/BN(1) model.
- (2) *Homogeneous models*:—The model constructed from the background dataset. Two types of background models are considered:
  - (a) Markov( $L$ )—the homogeneous Markov model of order  $L$ .
  - (b) VOM( $L, c$ )—the homogeneous VOM model with  $L$  denoting the initial maximal order and  $c$  denoting the pruning constant, which is equivalent to the Markov( $L$ ) model in case of  $c = 0$ .
- (3) *Mean TP*: The obtained mean TP rate for a fixed TN level of 99.9%. The standard deviation of the estimated mean TP (in Figs 4–6) equals  $S/\sqrt{10^6} \approx 0.01\%$ , where  $S$  denotes the sample standard deviation that is obtained from the  $10^6$  fold replicated stratified-holdout sampling experiment. The standard deviations of each experiment are not shown as they are similar.
- (4) *Number of nodes*: The average number of nodes in the model over the stratified-holdout experiments.

In the first stage of the experiment we study the classification performance of foreground inhomogeneous VOB models versus inhomogeneous Markov models, including the widely used PWM model (e.g. Fickett and Hatzigeorgiou, 1997; Ewens and Grant, 2001; Mount, 2001). We use the random background dataset and obtain a homogeneous Markov(0) as the best background model. We find that inhomogeneous Markov(2) and Markov(3) models are overfitted and that the inhomogeneous Markov(1) model achieves the highest classification performance of the fixed-order models with a



**Fig. 4.** Mean TP values for different fixed-order model combinations. The Markov model order and their number of nodes in the background model are presented in the abscissa. The three different curves signify the different foreground models: PWM, Markov(1) and a VOB(1,0)/BN(1) which is equivalent to a BT.



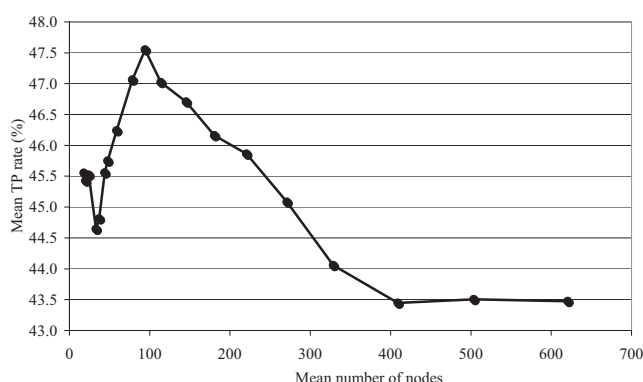
**Fig. 5.** Pruning of the foreground VOB(1,  $c$ ) model for different values of the pruning constant  $c$  for a Markov(3) background model. The number of nodes in the foreground model is presented on the abscissa to show the size of the model.

mean TP rate of 29.4%. Pruning the inhomogeneous VOM models by different pruning constants does not result in considerable improvements over fixed-order models: the VOB(1,  $c$ ) with  $c = 0.210$  achieves a statistically significant improvement with a mean TP rate of 30.7%.

Next, we analyse the performance of higher-order Markov models based on the Sigma-70 foreground dataset and the intergenic background dataset. We summarize the results in Figures 4–6. Figure 4 focuses on fixed-order models. It shows the performance of different combinations of foreground and background models. Figure 5 shows the enhancement from pruning the foreground model to obtain a VOB, where the background model is fixed to Markov(3). Figure 6 shows the improvement from pruning a 5th-order VOM model in the background with the foreground model fixed to the best VOB model from Figure 5. The experiment details now follows.

Figure 4 presents the classification accuracy for different combinations of fixed-order foreground (PWM, inhomogeneous Markov(1), and a BT) and background models (homogeneous Markov  $L = 0$  up to  $L = 5$ ). For a PWM model as the foreground model, a Markov(2)





**Fig. 6.** Pruning the background VOM(5,  $c$ ) model for different values of the pruning constant  $c$ . The foreground model is the VOB(1,  $2^{-3.75}$ ) model, which was found to be the best model in Figure 5 and has  $\sim 36$  nodes. The number of nodes in the background model is presented on the abscissa to show the size of the model.

model is the optimal background model, yielding a mean TP rate of 44.39%. We also find that, if an inhomogeneous Markov(1) model is chosen as foreground model, then a Markov(3) model is the optimal background model with a mean TP rate of 43.5%. Finally, with a BT model as the foreground model, a Markov(3) model is the optimal background model, resulting in the highest mean TP rate over all fixed-order models of 45.65%. All findings are in qualitative agreement with previous studies, such as, Thijs *et al.* (2001) and Barash *et al.* (2003), that indicate that homogeneous Markov(0) models may not be optimal for modeling genomic DNA. Clearly, the optimal Markov model order for the background depends on the foreground model chosen as well as on the measure of classification accuracy. In addition, we caution that one cannot infer from Figure 4 the optimal Markov model order for other datasets or other classification problems. One interesting observation from Figure 4 is that for all background models, the mean TP rates of the PWM as foreground models are higher than of the inhomogeneous Markov(1) model. Thus, for all background models tested, the PWM model is superior to the inhomogeneous Markov(1) model. This finding is consistent with the high popularity of the PWM model for TFBS recognition and explains why the weight array model proposed by Zhang and Marr (1993), which corresponds to an inhomogeneous Markov(1) model and which has been shown to be more accurate than the PWM model for splice site recognition, has not replaced the PWM model for TFBS recognition. The unimodal behaviour of the TP rate as a function of the number of model parameters might reflect the trade-off between over-fitted models and under-fitted models. It can also be seen from Figure 4 that for third and higher order background models, the BT/BN(1) outperforms the PWM model. For lower order background models, there is no clear dominance.

Although for certain background models, the BT models achieve better results than PWM models, it is not clear whether the BTs are overfitted. In the following, we test for such an overfitting, scrutinizing VOB(1,  $c$ ) models for different pruning constants since these models have the potential of modeling only a few (significant) statistical dependencies, while neglecting statistically insignificant ones.

Figure 5 presents the classification performance of different foreground VOB(1,  $c$ ) models for different choices of the pruning constant

$c$  (including the PWM model) for a background Markov(3) model, which was found to be the best model in Figure 4. The mean TP rate is given as a function of the mean number of nodes in the VOB(1,  $c$ ). Note that increasing the pruning constant decreases the mean number of nodes in the VOB(1,  $c$ ). The behavior of the mean TP rate is nearly unimodal except for models very similar to PWM model. The best foreground model is a VOB(1,  $2^{-3.75}$ ) achieving a mean TP rate of 46.46%. This is an improvement of 0.8% over the BT/BN(1) model. As one can see from Figure 5, the best VOB(1,  $c$ ) foreground model has approximately half the number of nodes compared to the full model, yet reaches a significantly higher mean TP rate.

To further explore the dependencies between foreground and background models, we now fix the foreground model to a VOB(1,  $2^{-3.75}$ ) and turn to homogeneous VOM models for the background. Figure 6 presents the improvement gained by pruning a background VOM model with maximum order 5. The plot has again an essential unimodal shape as in Figure 5. The best background model found is a VOM(5,  $2^{-5.5}$ ), which has 94 nodes (it is too large to be presented in the paper, but it is available upon request from the authors). This best combination of a foreground VOB(1,  $2^{-3.75}$ ) with a background VOM(5,  $2^{-5.5}$ ) achieves a mean TP rate of 47.56%. This is an improvement of 3.17% compared to the combination of PWM/Markov(2) models in Figure 4.

It is interesting to note that this mean TP rate can only be gained if we use a maximal order of 5 for the VOM. In the VOM(5,  $2^{-5.5}$ ) model, there are still contexts up to the 5th order (e.g. GCCGG, TCCGG), while others are already pruned down to order 3. These long contexts seem to be responsible for the high classification accuracy of the background model, as pruning a Markov model of initial order 4 or 3 reaches significantly lower mean TP rates.

As can be seen from Figure 6, for low pruning constants (models with more than 300 nodes) the mean TP rate is even below that of the combination of PWM and Markov(L). The VOM(5,  $c$ ) models between  $c = 2^{-10}$  and  $2^{-9}$  still have 400–600 nodes, which is more than for a 4th order Markov model (with 341 nodes) and causes a strong over-fitting effect. The best mean TP rate is reached for a pruned model (with 94 nodes) that has only a few more parameters than the fixed Markov(3) model, but utilizes those parameters more efficiently. If the pruning constant is further increased, the classification accuracy decreases again, as now significant contexts are pruned from the model and fixed-order Markov models.

To summarize, the variable-order concept applied to foreground Bayesian networks (obtaining the VOB(1,  $c$ ) model) and to background Markov models (obtaining the VOM model) is shown to outperform the PWM model.

Figure 3 shows the foreground VOB(1,  $2^{-3.75}$ ) model which reached the highest TP rate. Note that more than half the edges in the dependency graph are between non-adjacent positions, and that two of the edges are between positions from separate boxes (the ‘–35 box’ and ‘–10 box’). Approximately half of the edges were found to be insignificant and were pruned.

## 5 CONCLUSIONS

Variable-order Bayesian network models are one promising generalization of the widely used PWM model, fixed-order Markov models and BN models. In this paper we show that VOB(1,  $c$ ) models are useful for predicting the location of TFBSs. Specifically, we show in

stratified-holdout experiments that a VOB model can predict the location of sigma-70 binding sites in *E.coli* with higher accuracy than a PWM model, a fixed-order Markov model and a BN model. We speculate that VOB models might be useful for predicting the location of TFBSs in other genomes.

## ACKNOWLEDGEMENTS

We thank Hanspeter Herzel and Lev Levitin for valuable discussions and the German Ministry of Education and Research (BMBF Grant No. 0312706A), and the Minerva Foundation (Short Term Research Grant) for financial support.

## REFERENCES

- Baldi, P. and Brunak, S. (2001) *Bioinformatics: The Machine Learning Approach*, 2nd edn. The MIT Press, Cambridge, MA.
- Barash, Y., Elidan, G., Friedman, N. and Kaplan, T. (2003) Modeling dependencies in protein-DNA binding sites. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, NY, ACM.
- Bejerano, G. and Yona, G. (2001) Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, **17**, 23–43.
- Ben-Gal, I. and Shmilovici, A. (2001) Promoters recognition by variable-length Markov models. In *Workshop on Artificial Intelligence and Heuristic Methods for Bioinformatics*, San-Miniato, Italy.
- Ben-Gal, I., Morag, G. and Shmilovici, A. (2003) CSPC: a monitoring procedure for state dependent processes. *Technometrics*, **45**, 293–311.
- Benos, P.V., Lapides, A.S., Fields, D.S. and Stormo, G.D. (2001) SAMIE: statistical algorithm for modeling interaction energies. In *PSB'01*.
- Bilu, Y., Linial, M., Slonim, N. and Tishby, N. (2002) Locating transcription factors binding sites using a variable memory Markov model. *Leibnitz Center TR 2002-57*.
- Blattner, F.R. and Schroeder, J.L. (1984) A computer package for DNA sequence analysis. *Nucleic Acids Res.*, **12**, 615–617.
- Boutillier, C., Friedman, N., Goldszmidt, M. and Koller, D. (1996) Context-specific independence in Bayesian networks. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, August 1–4, 1996, Reed College, Portland, Oregon, USA, pp. 115–123. Also appears at <http://www.informatik.uni-trier.de/~ley/db/conf/uai/uai1996.html>
- Buhlmann, P. and Wyner, A.J. (1999) Variable length Markov chains. *Ann. Statist.*, **27**, 480–513.
- Bulyk, M.L. et al. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.*, **30**, 1255–1261.
- Cai, D. et al. (2000) Modeling splice sites with Bayes Networks. *Bioinformatics*, **16**, 152–158.
- Castelo, R. and Guigo, R. (2004) Splice site identification by idIBNs. *Bioinformatics*, **20**, i69–i76.
- Chow, C.K. and Liu, C.N. (1968) Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inform. Theory*, **14**, 462–467.
- Chu, S. et al. (1998) The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.
- Djordjevic, M. et al. (2003) A biophysical approach to transcription factor binding site discovery. *Genome Res.*, **13**, 2381–2390.
- Ewens, W.J. and Grant, G.R. (2001) *Statistical Methods in Bioinformatics: An Introduction*. Springer-Verlag, New York, Inc.
- Fickett, J.W. and Hatzigeorgiou, A.G. (1997) Eukaryotic promoter recognition. *Genome Res.*, **7**, 861–878.
- Friedman, N. and Goldszmidt, M. (1996) Learning Bayesian networks with local Structure. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, August 1–4, 1996, Reed College, Portland, Oregon, USA, pp. 252–262. Also appears at <http://www.informatik.uni-trier.de/~ley/db/conf/uai/uai1996.html>
- Friedman, N. et al. (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.
- Hansch, D. et al. (2002) Co-clustering of biological networks and gene expression data. *Bioinformatics*, **1**, 1–10.
- Heckerman, D., Geiger, D. and Chickering, D.M. (1995) Learning Bayesian networks. The combination of knowledge and statistical data. *Machine Learning*, **20**, 197–243.
- Hughes, J.D. et al. (2000) Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.*, **296**, 1205–1214.
- Kel, A.E. et al. (2003) MATCH: a tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.*, **31**, 3576–3579.
- Kel-Margoulis, O.V. et al. (2003) Composition-sensitive analysis of the human genome for regulatory signals. In *Silico Biol.*, **3**, 13.
- Kullback, S. (1959) *Information Theory and Statistics*. Wiley, New York (reprinted in 1978 by Peter Smith, MA).
- Liu, J.S. et al. (1995) Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Am. Stat. Assoc.*, **90**, 1156–1170.
- Liu, X. et al. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, **6**, 127–138.
- Mount, D.W. (2001) *Bioinformatics, Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, pp. 357–365.
- Neuwald, A.F. et al. (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci.*, **4**, 1618–1632.
- Ohler, U. and Niemann, H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends Genet.*, **17**, 56–60.
- Ohler, U. et al. (1999) Interpolated Markov chains for eukaryotic promoter recognition. *Bioinformatics*, **15**, 362–369.
- Orlov, Y.L. and Potapov, V.N. (2000) *Determining Markov Model of Genetical Texts by Stochastic Complexity Estimation*. BGRS, Novosibirsk, pp. 71–73.
- Orlov, Y.L. et al. (2002) Construction of stochastic context trees for genetic texts. In *Silico Biol.*, **2**, 233–247.
- Pickert, L. et al. (1998) Transcription regulatory region analysis using signal detection and fuzzy clustering. *Bioinformatics*, **14**, 244–251.
- Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, California.
- Rissanen, J. (1983) A universal data compression system. *IEEE Trans. Inform. Theory*, **29**, 656–664.
- Ron, D. et al. (1996) The power of amnesia: learning probabilistic automata with variable memory length. *Machine Learning*, **25**, 117–149.
- Salgado, H. et al. (2000) RegulonDB (version 3.0): transcriptional regulation and operon organization in *Escherichia coli* K-12. *Nucleic Acids Res.*, **28**, 65–67.
- Salzberg, S.L. (1997) A method for identifying splice sites and translational start sites in eukaryotic mRNA. *CABIOS*, **13**, 365–376.
- Salzberg, S.L. et al. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.
- Salzberg, S.L. et al. (1999) Interpolated Markov models for eukaryotic gene finding. *Genomics*, **59**, 24–31.
- Spellman, P.T. et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.
- Stormo, G.D. and Fields, D.S. (1998) Specificity, free energy and information content in protein-DNA interactions. *Trends Biochem. Sci.*, **23**, 109–113.
- Thijs, G. et al. (2001) A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, **17**, 1113–1122.
- Vert, J.P. (2001) Adaptive context trees and text clustering. *IEEE Trans. Inform. Theory*, **47**, 1884–1901.
- Weinberger, M. et al. (1995) A universal finite memory source. *IEEE Trans. Inform. Theory*, **41**, 643–652.
- Wingender, E. et al. (2000) TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Res.*, **28**, 316–319.
- Wingender, E. (2001) The TRANSFAC system on gene expression regulation. *Nucleic Acids Res.*, **29**, 281–283.
- Zhang, M.Q. and Marr, T.G. (1993) A weight array method for splicing signal analysis. *Comput. Appl. Biosci.*, **9**, 499–509.
- Zhao, X. et al. (2004) Finding Short DNA Motifs Using Permuted Markov Models. *Proceedings of the 8th Annual International Conference on Computational Molecular Biology*, 68–75, March 27–31, 2004, San Diego, California, USA, ACM.