JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 9, Number 2, 2002
© Mary Ann Liebert, Inc.
Pp. 447–464

A Gibbs Sampling Method to Detect Overrepresented Motifs in the Upstream Regions of Coexpressed Genes

GERT THIJS,[1] KATHLEEN MARCHAL,[1] MAGALI LESCOT,[1] STEPHANE ROMBAUTS,[2]
BART DE MOOR,[1] PIERRE ROUZÉ,[3] and YVES MOREAU[1]

ABSTRACT

Microarray experiments can reveal important information about transcriptional regulation. In our case, we look for potential promoter regulatory elements in the upstream region of coexpressed genes. Here we present two modifications of the original Gibbs sampling algorithm for motif finding (Lawrence *et al.*, 1993). First, we introduce the use of a probability distribution to estimate the number of copies of the motif in a sequence. Second, we describe the technical aspects of the incorporation of a higher-order background model whose application we discussed in Thijs *et al.* (2001). Our implementation is referred to as the Motif Sampler. We successfully validate our algorithm on several data sets. First, we show results for three sets of upstream sequences containing known motifs: 1) the G-box light-response element in plants, 2) elements involved in methionine response in *Saccharomyces cerevisiae*, and 3) the FNR O_2-responsive element in bacteria. We use these data sets to explain the influence of the parameters on the performance of our algorithm. Second, we show results for upstream sequences from four clusters of coexpressed genes identified in a microarray experiment on wounding in *Arabidopsis thaliana*. Several motifs could be matched to regulatory elements from plant defence pathways in our database of plant *cis*-acting regulatory elements (PlantCARE). Some other strong motifs do not have corresponding motifs in PlantCARE but are promising candidates for further analysis.

Key words: motif finding, Cribbs sampling, regulatory elements, gene expression, microarray.

1. INTRODUCTION

Microarrays let biologists monitor the mRNA expression levels of several thousands of genes in one experiment (for a review, see Lockhart and Winzeler [2000]). An interesting application of this microarray technology is to measure the evolution of mRNA levels at consecutive time points during

[1] ESAT-SCD, KULeuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium.
[2] Plant Genetics, VIB, University Gent, Ledeganckstraat 35, 9000 Gent, Belgium.
[3] INRA associated laboratory, VIB, University Gent, Ledeganckstraat 35, 9000 Gent, Belgium.

a biological experiment. This way, we can construct a temporal expression profile for the thousands of genes present on the array. Such large-scale microarray experiments have opened new research directions in transcript profiling (Sherlock, 2000; Szallasi, 1999; Wen *et al.*, 1998; Zhang, 1999). For instance, a primary goal in the analysis of such large data sets is to find genes that have similar behavior under the same experimental conditions. Several clustering algorithms are available to group genes that have a similar expression profile (Altman and Raychaudhuri, 2001; De Smet *et al.*, 2002; Eisen *et al.*, 1998; Heyer *et al.*, 1999; Mjolsness *et al.*, 2000; Tavazoie *et al.*, 1999). Given a cluster of genes with highly similar expression profiles, we can search for the mechanism that is responsible for their coordinated behavior. We basically assume that coexpression frequently arises from transcriptional coregulation. As coregulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, their promoter regions might contain some common motifs that are binding sites for transcription regulators. A sensible approach to detect these regulatory elements is to search for statistically overrepresented motifs in the promoter region of such a set of coexpressed genes (Bucher, 1999; Ohler and Niemann, 2001; Roth *et al.*, 1998; Tavazoie *et al.*, 1999; Zhu and Zhang, 2000).

Algorithms to find regulatory elements can be divided into two major classes: 1) methods based on word counting (Jensen and Knudsen, 2000; van Helden *et al.*, 1998, 2000a; Sinha and Tompa, 2000; Tompa, 1999; Vanet *et al.*, 2000) and 2) methods based on probabilistic sequence models (Bailey and Elkan, 1995; Hughes *et al.*, 2000; Lawrence *et al.*, 1993; Liu *et al.*, 2002; Roth *et al.*, 1998; Workman and Stormo, 2000). The word-counting methods analyze the frequency of oligonucleotides in the upstream region and use intelligent strategies to speed up counting and to detect significantly overrepresented motifs. These methods then compile a common motif by grouping similar words. Word counting methods lead to a global solution as compared to the probabilistic methods. The probabilistic methods represent the motif by a position probability matrix and the remainder of the sequence is modeled by a background model. To find the parameters of this model, these methods use maximum likelihood estimation in the form of Expectation Maximization (EM) and Gibbs sampling—EM is a deterministic algorithm and Gibbs Sampling is a stochastic equivalent of EM.

In this paper, we present two modifications of the original Gibbs sampling algorithm by Lawrence *et al.* (1993). First, a probabilistic framework is used to estimate the expected number of copies of a motif in a sequence. The proposed method resembles the method used in MEME (Bailey and Elkan, 1995). While Bailey and Elkan use a global variable to estimate the number of copies of the motif in the whole data set, we propose to estimate the number of copies of the motif in each sequence individually. For instance, when using AlignACE (Roth *et al.*, 1998) the user should estimate the total number of motif occurrences in the complete data set, while in our approach there is only a maximum number per sequence to be set. Second, we introduce the use of a higher-order background model based on an Markov chain. Here we comment on the technical details of the incorporation of the background model in the algorithm. The construction and influence of the background on the performance of the Motif Sampler has been described elsewhere (Thijs *et al.*, 2001). We showed that the use of a higher-order background model has a profound impact on the performance of the motif finding algorithm. In this manuscript, we focus on the details of the incorporation of these modifications in the Gibbs sampling algorithm to find the parameters of the extended probabilistic sequence model.

Our implementation of the Gibbs sampler was successfully tested on different data sets of intergenic sequences. We used data sets of upstream regions with known regulatory elements in *Arabidopsis thaliana*, in *Saccharomyces cerevisiae*, and in bacteria to demonstrate the performance of the Motif Sampler. In this manuscript, we discuss these results in detail and show how the different parameter settings influence the detection performance of the motif finding. Finally, we show some results for the upstream sequences of coexpressed genes identified in a microarray experiment on wounding in *A. thaliana* (Reymond *et al.*, 2000).

2. MODEL DESCRIPTION

In this section, we start with the basic description of the sequence model. Then we discuss the higher-order background model, and we introduce the probabilistic framework to estimate the number of copies of the motif in the sequence.

2.1. Basic sequence model

To start, we introduce the basic model that we use to represent a DNA sequence. The basic model assumes that one or more motifs are hidden in a noisy background sequence. On the one hand, the motif model is based on a frequency residue model (Lawrence *et al.*, 1993; Bailey and Elkan, 1995) and is represented by a position probability matrix θ_W:

$$\text{Motif } \theta_W = \begin{pmatrix} q_1^A & q_2^A & \cdots & q_W^A \\ q_1^C & q_2^C & \cdots & q_W^C \\ q_1^G & q_2^G & \cdots & q_W^G \\ q_1^T & q_2^T & \cdots & q_W^T \end{pmatrix},$$

with W the fixed length of the motif. Each entry q_i^b in the matrix θ_W gives the probability of finding nucleotide b at position i in the motif. On the other hand, the background model is represented by a transition matrix B_m, with m the order of the model (see Section 2.2). The probability P_0 that the sequence S is generated by the background model B_m is given by

$$P_0 = P(S|B_m) = P(b_1) \prod_{l=1}^{L} P(b_l|b_{l-1} \ldots b_1, B_m),$$

where b_l is the nucleotide at position l in the sequence S and $P(b_l|b_{l-1} \ldots b_1, B_m)$ is the probability of finding the nucleotide b_l at that position l in the sequence according to the background model and the sequence content. If the order of the background model is set to zero, the background model is represented by the single nucleotide model P_{snf} or the residue frequencies in the data set:

$$P_{\text{snf}} = [q_0^A \quad q_0^C \quad q_0^G \quad q_0^T]^T.$$

Now that we have defined the parameters of the models, we can use these parameters to compute the probability of the sequence when the position of the motif is known. If the start position a of a motif θ_W is known, then the probability that the sequence is generated given the model parameters is

$$P(S|a, \theta_W, B_0) = \prod_{l=1}^{a-1} q_0^{b_l} \prod_{l=a}^{a+W-1} q_{l-a+1}^{b_l} \prod_{l=a+W}^{L} q_0^{b_l}, \tag{1}$$

where $q_{l-a+1}^{b_l}$ is the corresponding entry $(b_l, l-a+1)$ in the motif model θ_W.

2.2. Higher-order background model

The first extension to the original Gibbs sampling algorithm for motif finding (Lawrence *et al.*, 1993) we implemented is the use of a higher-order background model. An elaborate evaluation and discussion on the influence of a higher-order background model on motif detection has been described elsewhere (Thijs *et al.*, 2001). Here we will only summarize the issues relevant to the remainder of the article.

The most frequently cited algorithms using the probabilistic motif model, AlignACE (Hughes *et al.*, 2000) and MEME (Bailey and Elkan, 1995), use the single nucleotide frequency distribution of the input sequences to describe the background model. However, if we look more closely at state-of-the-art gene detection software, Glimmer (Delcher *et al.*, 1999), HMMgene (Krogh, 1997), and GeneMark.hmm (Lukashin and Borodowsky, 1998), all of them use higher-order Markov processes to model coding and noncoding sequences. Markov models have been introduced to predict eukaryotic promoter regions (Audic and Claverie, 1997) and recently this method was refined to interpolated Markov chains (Ohler *et al.*, 1999). Recently, higher-order background models have also been introduced in word-counting methods (Sinha and Tompa, 2000). In parallel with our research, others (Liu *et al.*, 2001; McCue *et al.*, 2001; Workman and Stormo, 2000) have suggested the use of related higher-order background models to improve their motif-detection algorithms.

Starting from the ideas incorporated in these gene and promoter prediction algorithms, we developed a background model based on a Markov process of order m. This means that the probability of the nucleotide b_l at position l in the sequence depends on the m previous bases in the sequence, and the factor $P(b_{l-1}\ldots b_1, b_l|B_m)$ simplifies to $P(b_l|b_{l-1}\ldots b_{l-m}, B_m)$. Such a model is described by a transition matrix. Given a background model of order m, we write the probability of the sequence S being generated by the background model as

$$P(S|B_m) = P(b_1, \ldots, b_m|B_m) \prod_{l=m+1}^{L} P(b_l|b_{l-1}, \ldots, b_{l-m}, B_m).$$

The probability $P(b_1, \ldots, b_m|B_m)$ accounts for the first m bases in the sequences, while $P(b_l|b_{l-1}, \ldots, b_{l-m}, B_m)$ corresponds to an entry in the transition matrix that comes with the background model B_m.

Important to know is that the background model can be constructed either from the original sequence data or from an independent data set. The latter approach is more sensible if the independent data set is carefully created, which means that the sequences in the training set come from only the intergenic region and thus do not overlap with coding sequences. Currently, we have constructed an independent background model for *Arabidopsis thaliana* (based on the sequences in Araset [Pavy *et al.*, 1999] and PlantGene [Thijs *et al.*, 2001]) and also for *Saccharomyces cerevisiae* (*www.ucmb.ulb.ac.be/bioinformatics/rsa-tools/*). Background models for other organism are under construction. Nevertheless, the algorithm can also be used for other organisms by building the background model from the input sequences.

2.3. Finding the number of occurrences of a motif

The clustering of gene expression profiles of a microarray experiment gives several groups of coexpressed genes. The basic assumption states that coexpression indicates coregulation, but we expect that only a subset of the coexpressed genes are actually coregulated. When searching for possible regulatory elements in such a set of sequences, we should take into account that the motif will appear only in a subset of the original data set. We therefore want to develop an algorithm that distinguishes between the sequences in which the motif is present and those in which it is absent. Furthermore, in higher organisms, regulatory elements can have several copies to increase the effect of the transcriptional binding factor in the transcriptional regulation. Figure 1 gives a schematic representation of this kind of data set. To incorporate these facts, we reformulate the probabilistic sequence model in such a way that we can estimate the number of copies of the motif in each sequence.

First we introduce a new variable Q_k, which is the number of copies of the motif θ_W in the sequence S_k and which is missing from our observations. Together with this variable Q_k, we also introduce the probability $\gamma_k(c)$ of finding c copies of the motif θ_W in the sequence S_k, with

$$\gamma_k(c) = P(Q_k = c|S_k, \theta_W, B_m). \tag{2}$$

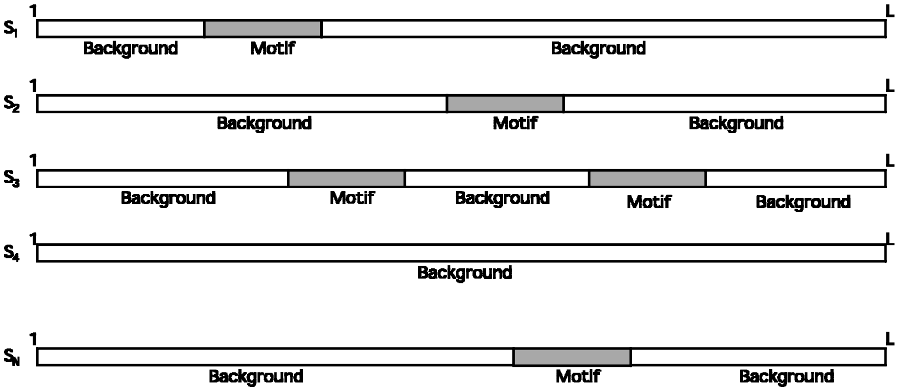

FIG. 1. Schematic representation of the upstream region of a set of co-expressed genes.

Equation 2 can be further expanded by applying Bayes' theorem:

$$\gamma_k(c) = \frac{P(S_k|Q_k = c, \theta_W, B_m)P(Q_k = c|\theta_W, B_m)}{P(S_k|\theta_W, B_m)}. \tag{3}$$

We can distinguish three different parts in Equation 3. The first part $P(S_k|Q_k = c, \theta_W, B_m)$ is the probability that the sequence is generated by the motif model θ_W, the background model B_m, and the given number of copies c. This probability can be calculated by summing over all possible combinations of c motifs,

$$\begin{aligned} &P(S_k|Q_k = c, \theta_W, B_m) \\ &\quad = \sum_{a_1} \cdots \sum_{a_c} P(S_k|a_1, \ldots, a_c, Q_k = c, \theta_W, B_m)P(a_1, \ldots, a_c|Q_k = c, \theta_W, B_m), \end{aligned} \tag{4}$$

where $a_1, \ldots, a_c$ are the start positions of the different copies of the motif. We assume that the prior $P(a_1, \ldots, a_c|Q_k = c, \theta_W, B_m)$ is independent of the motif model and of the background model and that it is therefore a constant inverse proportional to the total number of possible combinations. The probability $P(S_k|a_1, \ldots, a_c, Q_k = c, \theta_W, B_m)$ can be easily calculated with Equation 1. Note, however, that the complexity of the computations depends on the number of copies c, since all possible combinations of c motifs in the sequence have to be taken into account.

The second part in Equation 3 is the prior $P(Q_k = c|\theta_W, B_m)$, the probability of finding c copies given the motif model and the background model. For simplicity reasons, we assume that this probability is independent of θ_W and B_m and therefore it can be estimated as $P(Q_k = c)$. To efficiently calculate Equation 4, an adapted version of the *forward* algorithm can be used.

The final part is the probability $P(S_k|\theta_W, B_m)$. This probability can be calculated by taking the sum over all possible numbers of copies

$$P(S_k|\theta_W, B_m) = \sum_{c=0}^{\infty} P(S_k|Q_k = c, \theta_W, B_m)P(Q_k = c|\theta_W, B_m). \tag{5}$$

From a more practical point of view, this equation is not workable. Taking the sum over all possible numbers of copies is impractical. Therefore, we introduced a new parameter $C_{\max}$ to set the maximal number of copies expected in each sequence. The sum in Equation 5 is substituted with a sum going from 0 to $C_{\max}$. The influence of this parameter $C_{\max}$ on the performance of the algorithm will be discussed in detail in Section 5. If we have computed $\gamma_k(c)$, for $c = 1, \ldots, C_{\max}$, this probability can be used to estimate the expected number of copies Q_k of the motif in the given sequence S_k,

$$\begin{aligned} E_{(S_k,\theta_W,B_m)}(Q_k) &= \sum_{c=0}^{C_{\max}} cP(Q_k = c|S_k, \theta_W, B_m) \\ &= \sum_{c=1}^{C_{\max}} c\gamma_k(c). \end{aligned} \tag{6}$$

3. ALGORITHM AND IMPLEMENTATION

In the previous section, we discussed the technicalities of the higher-order background model and the estimation of the number of copies of a motif in the sequences. In this section, we describe the incorporation of the presented modifications in the iterative procedure of the Gibbs sampling algorithm. First we describe the algorithm in detail. We do not give a general description of the Gibbs sampling methodology, as it is available elsewhere (Lawrence *et al.*, 1993; Liu *et al.*, 2002) but we focus on the description of our implementation.

In the following paragraph, we briefly address the problem of finding different motifs. Then we give an overview of the output returned by the Motif Sampler and the different motif scores. Finally, we discuss the web interface to the Motif Sampler, and we give a brief description of all the parameters that the user has to define.

3.1. Algorithm

The input of the Motif Sampler is a set of upstream sequences. In the first step of the algorithm, the higher-order background model is chosen. The background model can be pre-compiled or it can be calculated from the input sequences. The algorithm then uses this background model B_m to compute, for each segment $x = \{b_l, b_{l+1}, \ldots, b_{l+W-1}\}$ of length W in every sequence, the probability

$$P_{\text{bg}}(x) = P(x|S, B_m) = \prod_{i=0}^{W} P(b_{l+i}|b_{l+i-1}\ldots b_{l+i-m}, B_m)$$

that the segment was generated by the background model. These values are stored and there is no need to update them during the rest of the algorithm.

In the second step, for each sequence S_k, $k = 1, \ldots, N$, the alignment vector $A_k = \{a_{k,c}|c = 1, \ldots, C_{\max}\}$ is initialized from a uniform distribution. The number of copies Q_k is sampled according to the initial distribution Γ_k, with

$$\Gamma_k = \{\gamma_k(c)|c = 0, \ldots, C_{\max}\}.$$

In the next step, the algorithm loops over all sequences, and the alignment vector for each sequence is updated. First, the motif model $\tilde{\theta}_W$ is calculated based on the current alignment vector. This estimated motif model is used to compute the probability distribution $W_z(x)$ over the possible motif positions in sequence S_z. The calculation of $W_z(x)$ is similar to the predictive update formula as described by Liu *et al.* (1995), but we substituted the single nucleotide background model with a higher-order background model, which leads to the following equation

$$W_z(x) = \frac{P(x|\tilde{\theta}_W)}{P(x|S, B_m)} = \prod_{i=0}^{W-1} \frac{\tilde{\theta}_W(i+1, b_{l+i})}{P(b_{l+i}|b_{l+i-1}\ldots b_{l+i-m}, B_m)}. \tag{7}$$

Next, a new alignment vector is selected by taking $C_{\max}$ samples from the normalized distribution $W_z(x)$. Given the estimated motif model $\tilde{\theta}_W$, the algorithm reestimates the distribution Γ. Although we sampled $C_{\max}$ positions, only the first Q_k positions will be selected to form the final motif. This procedure is ended when there is no more improvement in the motif model or we exceed a maximum number of iterations.

1. Select or compute the background model B_m.
2. Compute the probability $P_{\text{bg}}(x)$ for all segments x of length W in every sequence.
3. Initialize of the alignment vectors $A = \{A_k|k = 1\ldots N\}$ and the weighting factors $\Gamma = \{\Gamma_k|k = 1\ldots N\}$.
4. Sample Q_k, for each sequence S_k, from the corresponding distribution Γ_k.
5. For each sequence S_z, $z = 1, \ldots, N$:
   a. Create subsets $\tilde{S} = \{S_i|i \neq z\}$ and $\tilde{A} = \{\tilde{A}_i|i \neq z\}$, with $\tilde{A}_i = \{a_{i,1}, \ldots, a_{i,Q_i}\}$.
   b. Calculate $\tilde{\theta}_W$ from the segments indicated by the alignment vectors $\tilde{A}$.
   c. Assign to each segment x in S_z the weight $W_z(x)$ conforming to Equation 7.
   d. Sample $C_{\max}$ alignment positions to create the new vector A_z from the normalized distribution W_z.
   e. Update the distribution Γ_z.
6. Repeat from Step 4 until convergence or for a maximal number of iterations.

3.2. Inclusion of the complementary strand

Often it is also useful to include the complementary strand into the analysis procedure since transcription binding factors are known to bind on both strands of the DNA. The straightforward way to tackle this problem is to double the size of the data set by including the reverse complement of each individual sequence. However, with this approach, the noise present in the data set will also be doubled. Therefore, we suggest a more careful approach. In Step 5 of the algorithm, the predictive update distribution of both the sequence S_z and its reverse complement is computed. Next, the alignment positions are sampled from S_z, these positions are masked on the opposite strand, and the alignment positions on the reverse complementary sequence are sampled. Finally, the distribution Γ_z is calculated and updated for both the strands.

3.3. *Finding multiple motifs*

So far, we discussed only the issue of finding one motif that can have multiple copies, but we would like also to find multiple motifs. To find more than one motif, we will run the Motif Sampler several times, and in each run we will mask the positions of the motifs previously found. By masking the positions, it will be impossible to find the same motif twice. Masking a certain position in the sequence can be achieved by forcing the weights $W_z(x)$ to be 0 for all segments x that overlap with the previous motifs. The allowed overlap is a parameter that the user of the algorithm has to define.

3.4. *Motif scores*

The final result of the Motif Sampler consists of three parts: the position probability matrix θ_W, the alignment vector A, and the weighting factors Γ. Based upon these values, different scores with their own characteristics can be calculated: consensus score, information content, and log-likelihood.

The consensus score is a measure for the conservation of the motif. A perfectly conserved motif has a score equal to 2, while a motif with a uniform distribution has a score equal to 0.

$$\text{Consensus Score} = 2 - \frac{1}{W}\sum_{l=1}^{W}\sum_{b\in\{\text{A,C,G,T}\}} q_l^b \log(q_l^b) \tag{8}$$

The information content or Kullback-Leiber distance between the motif and the single nucleotide frequency tells how much the motif differs from the single nucleotide distribution. This score is maximal if the motif is well conserved and differs considerably from the background distribution.

$$\text{Information Content} = \frac{1}{W}\sum_{l=1}^{W}\sum_{b\in\{\text{A,C,G,T}\}} q_l^b \log\left(\frac{q_l^b}{q_0^b}\right). \tag{9}$$

As a final score, we consider the likelihood, $P(S, A|\theta_W, B_m)$, or the corresponding log-likelihood. The motif and alignment positions are the results of maximum likelihood estimation and therefore the log-likelihood is a good measure for the quality of the motif. In this case, we are especially interested in the positive contribution of the motif to the global log-likelihood. If we write the probability of the sequence being generated by the background model, $P(S|B_m)$, as P_0, the log-likelihood can be calculated as

$$\begin{aligned}\log\big(\pi(S, A|\theta_W, B_m)\big) &= \log\Big(\sum_{c=0}^{C_{\max}} \gamma_c P(S|A^c, \theta_W, B_m)P(A^c|\theta_W, B_m)\Big)\\ &= \log(C) + \log\Big(\sum_{c=0}^{C_{\max}} \gamma_c P(S|A^c, \theta_W, B_m)\Big),\end{aligned}$$

where A^c refers to the first c positions in the alignment vector A, which contains $C_{\max}$ positions, and C is a constant representing the prior probability of the alignment vector, which is assumed to be uniform. The log-likelihood depends on the strength of the motif and also on the total number of instances of the motif. Each of these scores accounts for a specific aspect of the motif. Together with these scores, the number of occurrences of a motif in the input sequences can be computed. We can use Equation 6 to estimate the number of occurrences of a motif in the data set.

3.5. Web interface

The implementation of our motif-finding algorithm is part of our INCLUSive web site (Thijs *et al.*, 2002) and is accessible through a web interface: *www.esat.kuleuven.ac.be/~dna/BioI/Software.html.* On this web interface, the user can either paste the sequence in FASTA format or upload a file to enter the sequences. The user also has to define five parameters. Here is a short description of these parameters.

- Background model B_m: Selects one of the precompiled models (*A. thaliana* or *S. cerevisiae*) as the background model or compute the background model from the sequence data themselves.
- Length W: Determines the length of the motif, which is fixed during the sampling. Reasonable values range from 5 to 15.
- Motifs N: Sets the number of different motifs to be searched for. The motifs will be searched for in consecutive runs while the positions of the previously found motifs are masked.
- Copies $C_{\max}$: Sets the maximum number of copies of a motif in every sequence. If this number is set too high, noise will be introduced in the motif model and the performance will degrade.
- Overlap O: Defines the overlap allowed between the different motifs. This parameter is used only in the masking procedure.

4. DATA

4.1. G-box sequences

To validate our motif-finding algorithm, we first constructed two data sets of gene upstream regions: 1) sequences with a known regulatory element, G-box, involved in light regulation in plants and 2) a selection of upstream sequences from *A. thaliana* in which no G-box is reported (random data set). The G-box data set consists of 33 sequences selected from PlantCARE (Lescot *et al.*, 2002) containing 500bp upstream of the translation start and in which the position of the G-box is reported. This data set is well suited to give a proof of concept and to test the performance of the Motif Sampler, since we exactly know the consensus of the motif `CACGTG` and also the known occurrences of the motif in the sequences. The data can be found at *www.plantgenetics.rug.ac.be/bioinformatics/lescot/Datasets/ListG-boxes.html.*

The random set consists of 87 sequences of 500bp upstream of translation start in genes from *A. thaliana.* The genes in this set are supposed to contain no G-box binding site. This set is used to introduce noise into the test sets and to evaluate the performance of the Motif Sampler under noisy conditions.

4.2. MET sequences

A set of upstream sequences from 11 genes in *S. Cerevisiae* that are regulated by the Cbfl-Met4p-Met28p complex and Met31p or Met32p in response to methionine (van Helden *et al.*, 1998) were obtained from *www.ucmb.ulb.ac.be/bioinformatics/rsa-tools/.* Upstream regions between −800 and −1 are selected. The consensus of both binding sites is given by `TCACGTG` for the Cbfl-Met4p-Met28p complex and `AAAACTGTGG` for Met31p or Met32p (van Helden *et al.*, 1998). Figure 2 shows the putative locations based on sequence homology of the two binding sites. These locations are used to validate the motifs retrieved with the Motif Sampler.

4.3. Bacterial sequences

As a third test set, we created a data set with intergenic sequences from bacteria. The data set contains a subset of bacterial genes regulated by the O_2-responsive transcriptional regulator FNR (Marchal, 1999). The genes were selected from several bacterial species: *Azospirillum brasilense*, *Paracoccus denitrificans*, *Rhodobacter sphaeroides*, *Rhodobacter capsulatus*, *Sinorhizobium meliloti*, and *Escherichia coli.* The data set contains 10 intergenic sequences of varying length. The FNR motif is described in the literature as an interrupted palindrome of 14bp and the consensus, `TTGACnnnnATCAA`, consists of two conserved blocks of 5bp separated by a fixed spacer of 4bp (as will be shown later, in Figure 6). The presence of the spacer will make the final motif model more degenerate and will have an influence on the performance of the Motif Sampler.

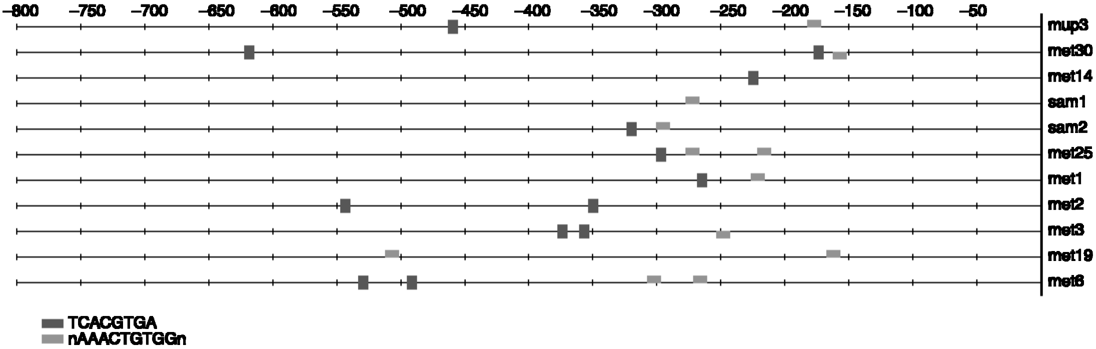

FIG. 2. Representation of the putative location of TCACGTG and AAACTGTGG in the MET sequence set.

4.4. Plant wounding data

As a last test set, we selected sequences from clusters of coexpressed genes. As a test case, we selected the data from Reymond *et al.* (2000), who measured the gene expression in response to mechanical wounding in *A. thaliana.* The mRNA was extracted from leaves at 30 minutes, 1 hour, 90 minutes, and 3, 6, 9, and 24 hours after wounding (7 time-points), and the expression level was measured on a cDNA microarray of 138 genes related to the plant defence mechanism. To find the groups of coexpressed genes, we use an adaptive quality-based clustering which we developed in our group (De Smet *et al.*, 2002) to identify groups of tightly coexpressed genes. This resulted in eight small clusters of coexpressed genes.

From the eight clusters found, four of them contained only three genes. These sets were not considered for further analysis. The remaining clusters contained, respectively, 11, 6, 5, and 5 sequences. The profiles of the clusters and the genes belonging to the clusters can be found at *www.plantgenetics.rug.ac.be/bioinformatics/lescot/Datasets/Wounding/Clusters.html.* The automated upstream sequence retrieval of INCLUSive (Thijs *et al.*, 2002) was used to find the upstream sequences of the genes in each of the selected clusters. Finally, we truncated each retrieved sequence to 500bp upstream of the annotated translation start.

5. RESULTS

5.1. G-box sequences

We exhaustively tested the performance of our implementation of the Motif Sampler. At first, we set up a test with only the 33 G-box sequences. In this test, we looked only at the positive strand. We searched for six different motifs of 8bp and the maximal number of copies is varied between 1 and 4 and the allowed overlap was fixed on 1bp. Each test was repeated 20 times. Table 1 gives an overview of the different scores. The first column indicates the maximal number of copies of the motif. The second column gives the number of runs in which the G-box consensus CACGTG was found. The next three columns give the average of the different scores: consensus score, information content, and log-likelihood (see Section 3.4). The last column shows the average number of sites in the data set that the algorithm identifies as being representatives of the motif.

Table 1. Average and Variance of the Scores of the Motif Found in the G-Box Data Set

Number of copies	*G-box found in 20 runs*	*Consensus score*	*Information content*	*Loglikelihood*	*Total occurrences*
1	18/20	1.56 ± 0.08	1.75 ± 0.11	268.02 ± 41.03	26.2 ± 3.1
2	15/20	1.34 ± 0.06	1.49 ± 0.09	260.98 ± 27.65	43.2 ± 4.3
3	18/20	1.30 ± 0.08	1.45 ± 0.08	249.75 ± 25.09	49.0 ± 8.9
4	15/20	1.30 ± 0.07	1.44 ± 0.07	255.76 ± 16.08	51.9 ± 9.2

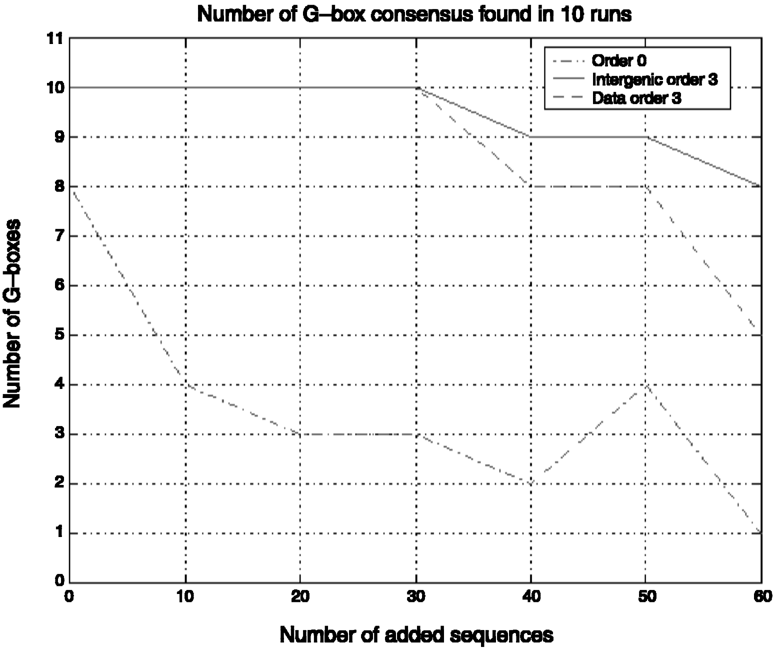

FIG. 3. Total number of times the G-box consensus is found in 10 repeated runs of the tests for three different background models. The data set consists of the 33 G-box sequences and a fixed number of added noisy sequences.

When the number of copies is set to 1, a more conserved motif is found, but a number of true occurrences is missed. Increasing the number of copies allows one to estimate better the true number of copies of the motif, but more noise is introduced into the initial model and the final model is more degenerate. This is clearly indicated by the consensus score and the information content. Both scores decrease as the maximal number of copies increases. The number of representatives of the motifs detected by the algorithm increases with the number of copies. We can also see that the difference is most pronounced when increasing the number of copies from 1 to 2, but that there is not much difference between 3 and 4 copies. The trade-off between the number of occurrences and the degeneracy of the motif has to be taken into account when trying to find the optimal parameters. For instance, when searching for 1 copy, the algorithm returns as consensus sequences `mCACGTGG` or `CCACGTGk`. When allowing up to 4 copies the algorithm will return more degenerate consensus sequences as `nnCACGTG` or `mCArGTGk`.

Another important issue is the influence of noise on the performance of the Motif Sampler. Noise is due to the presence of upstream sequences that do not contain the motif. To introduce noise in the data set, we added in several consecutive tests each time 10 extra sequences, in which no G-box is reported, to the G-box data set. We exhaustively tested several configurations to see how the noise influences the performance of the Motif Sampler. In this extensive test, we limited the number of repeats to 10 for each test corresponding to a different set of parameters. Figure 3 shows the total number of times the G-box consensus was detected in ten runs for three different background models and an increasing number of added sequences when searching for a motif of length 8bp and the maximal number of copies set to 1. As can be expected, the number of times the G-box is detected decreases when more noise is added to the original set of 33 G-box sequences. This influence is more dramatic for the single nucleotide background model than for the third-order background model. More details on the use of the higher-order background models are given elsewhere (Thijs *et al.*, 2001).

5.2. MET sequences

In a subsequent set of tests, we experimented with the MET sequence set (in this case both strands are analyzed). In these tests, we used the higher-order background models compiled from the upstream regions of all the annotated genes in the yeast genome (van Helden *et al.*, 2000b). With this data set, we further explored the influence of the different parameters on the performance of the algorithm. Preliminary tests showed that using a third-order background model was the best choice (data not shown), and we therefore used only the third-order background model in these tests. As in the G-box example, we ran our Motif Sampler with different combinations of parameters. In this case, we looked at the motif length W, the maximal number of copies $C_{\max}$, and the influence of the total number of motifs N. Three different motif lengths W were tested: 8, 10, and 12bp. The maximal number of copies $C_{\max}$ could vary between 1 and 4.

First, N was set to 1, and each test, with a given set of parameters, was repeated 100 times. Since there are two different binding sites present in this data set, the algorithm should be able to pick up both of them. When searching for only one motif, the algorithm is surely not able to capture both in one run. Therefore the algorithm was tested with the number of different motifs N also set to 2, 3, and 4. Again, each test with a particular set of parameters was repeated 100 times. This means that in each test an additional 100 motifs are found.

To analyze these results, we first looked at the information content of the all motifs found (Equation 9). Figure 4 shows how the information content changes when $C_{\max}$ is increased from 1 to 4. On the x-axis, the motifs are ranked with increasing information content, and the y-axis indicates the information content. Figure 4 clearly shows that when the maximal number of copies is increased the information content decreases. This is logical since more instances can be selected and the motif models become more degenerate. The effect is most pronounced when going from maximal 1 copy to 2 copies. Figure 5 shows how the information content changes when $C_{\max}$ is fixed at 1 but the motif length is set to 8, 10, or 12bp. Again, the plot clearly shows that on average the information content decreases when the length of the motif grows. This can be explained by the fact that the given information content is the average over all positions (see Equation 9). When the length of the motif is increased, noninformative positions can be added to the motif model, and therefore the information content decreases on average. This effect has to be taken into account when comparing the results of tests with different parameter settings.

Next, the found motifs are further analyzed. Since prior knowledge about the binding sites was available, it was possible to compare the alignment vector of each retrieved motif with the putative location of the binding site. For each motif, the positions of all instances of the motif were compared with these putative positions. If at least 50% of the alignment positions correspond with one of the two known sites, the motif was considered as a good representative. A threshold of 50% corresponds to the number of instances needed to create a motif that resembles the consensus of a true motif. Table 2 gives an overview of the results when looking for the motifs that sufficiently overlap with the putative motifs. This table shows the

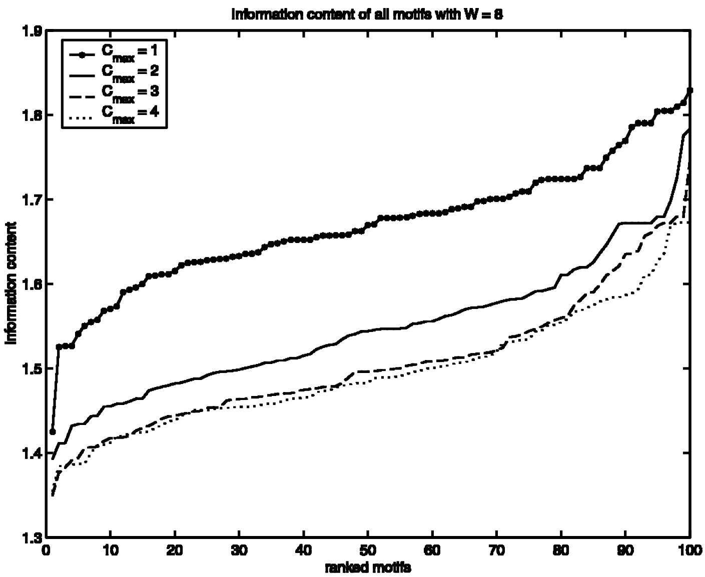

FIG. 4. Comparison of the information content of all motifs retrieved when searching for motifs of length 8bp and varying the number of maximal copies from 1 to 4.

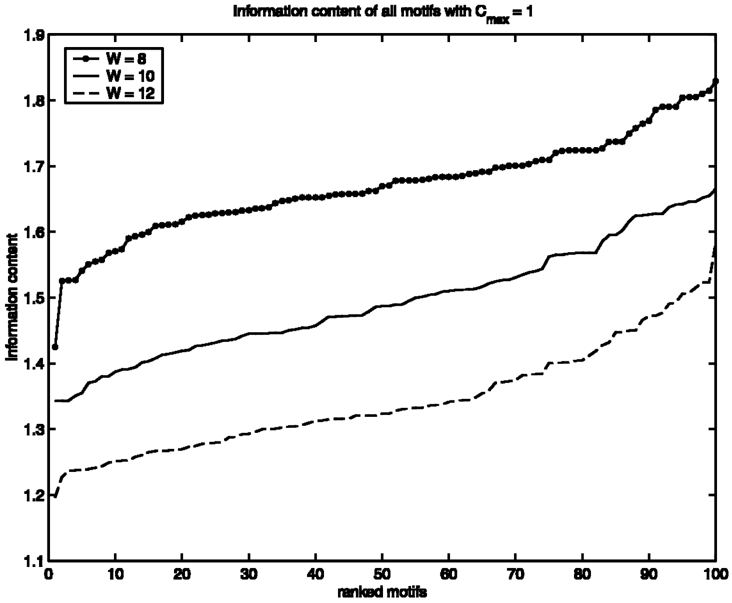

FIG. 5. Comparison of the information content of all motifs retrieved when searching for motifs of length 8, 10, and 12bp and the number of maximal copies fixed to 1.

number of motifs found that overlap for three different motif lengths and four different values of $C_{\max}$. The top part shows the result when we look at only one motif per run (100 motifs); the bottom part shows the results when we searched for four different motifs per run (400 motifs). When looking at the known motif TCACGTG, we can clearly see that the algorithm performed best when we searched for a motif of length 8bp and maximum 1 copy. This motif is only found in a few runs as the first motif when searching for motifs of length 12; however, the performance improves when the number of motifs is increased to four per run. If we look at the motif AAACTGTGG, the overall performance is rather similar for all parameter settings. There is almost no increase of correct motifs as the number of motifs increases to four per run.

Another way to analyze the results is by looking at the retrieved motifs with the highest scores. This approach allowed us to check whether the best motifs found corresponded to the true motifs. In this particular case, we looked at the information content (Equation 9). First, the motif with the highest information content was selected. Next, we removed all motifs in the list that were similar to the selected motif. Two motifs, θ_1 and θ_2, are considered similar if the mutual information of both motifs was smaller than 0.7. To

Table 2. Number of Times a Known Motif is Found in the MET Sequence in 100 Motifs (Top) and 400 Motifs (Bottom)

$C_{\max}$	$W = 8$	$W = 10$	$W = 12$	$W = 8$	$W = 10$	$W = 12$
1	33	10	4	16	14	8
2	12	4	6	17	19	15
3	14	3	2	17	24	8
4	11	1	3	11	11	12
1	59	58	49	19	16	11
2	50	59	35	17	20	15
3	26	23	8	17	25	8
4	26	10	9	11	11	12
		TCACGTG			AAAACTGTGG	

TABLE 3. THE SIX BEST SCORING MOTIFS OF LENGTH 8bp WHEN SEARCHING FOR 1, 2, 3 AND 4 DIFFERENT MOTIFS IN 100 TESTS

	100 motifs			*200 motifs*		
1	AACTGTGG	22	1.74	AACTGTGG	27	1.72
2	kCACGTGA	34	1.67	kCACGTGA	56	1.65
3	CGAAACCG	4	1.67	CGAAACCG	5	1.65
4	CGGsACCC	2	1.68	CGGsACCC	7	1.54
5	CTCCGGGT	3	1.69	AmGCCACA	4	1.66
6	CmGTCAAG	2	1.66	CTCCGGGT	5	1.67
	300 motifs			*400 motifs*		
1	AACTGTGG	28	1.71	AACTGTGG	35	1.66
2	kCACGTGA	62	1.64	kCACGTGA	64	1.63
3	CGAAACCG	7	1.61	CGAAACCG	7	1.61
4	CGGsACCC	10	1.54	CGGsACCC	15	1.50
5	AmGCCACA	7	1.61	AmGCCACA	7	1.60
6	CTCCGGGT	7	1.61	CTCCGGGT	7	1.61

calculate the mutual information, a shifted version of the motif is taken into account. To compensate for the length of the shifted motif, a normalizing factor was introduced in the formula of the mutual information, which leads to the following equation:

$$\text{mutual information} = \frac{1}{W}\sum_{j=1}^{W}\sum_{i=1}^{4}\theta_1(i,j) * \log_2\left(\frac{\theta_1(i,j)}{\theta_2(i,j)}\right). \tag{10}$$

After removal of the similar motifs, the same procedure was repeated six times on the reduced set of motifs. We applied this methodology on all different tests, but here we show the results for two particular parameter settings. Table 3 and Table 4 give an overview of the six best motifs found when searching for motifs of, respectively, 8bp and 12bp that can have maximum one copy. The tables are split in four

TABLE 4. THE SIX BEST SCORING MOTIFS OF LENGTH 12bp WHEN SEARCHING FOR 1, 2, 3 AND 4 DIFFERENT MOTIFS IN 100 TESTS

	100 motifs			*200 motifs*		
1	GAGGGCGTGTGC	4	1.48	GAGGGCGTGTGC	6	1.45
2	AAACTGTGGyGk	6	1.49	AAACTGTGGyGk	6	1.49
3	kAGTCAAGGsGC	2	1.38	kAGTCAAGGsGC	3	1.34
4	TGGTAGTCATCG	4	1.41	TGGTAGTCATCG	5	1.38
5	TrTGkrTGkGwG	5	1.36	TrTGkrTGkGwG	9	1.35
6	ACwGTGGCkTyn	4	1.35	ACwGTGGCkTyn	5	1.35
	300 motifs			*400 motifs*		
1	GAGGGCGTGTGC	7	1.42	GAGGGCGTGTGC	7	1.42
2	TGCTsCCAACnG	2	1.39	TGCTsCCAACnG	2	1.39
3	AAACTGTGGyGk	7	1.46	AAACTGTGGyGk	7	1.46
4	kAGTCAAGGsGC	3	1.34	kAGTCAAGGsGC	5	1.26
5	CCGGAGTCAGGG	2	1.36	CCGGAGTCAGGG	2	1.36
6	TGGTAGTCATCG	5	1.38	TGGTAGTCATCG	6	1.35

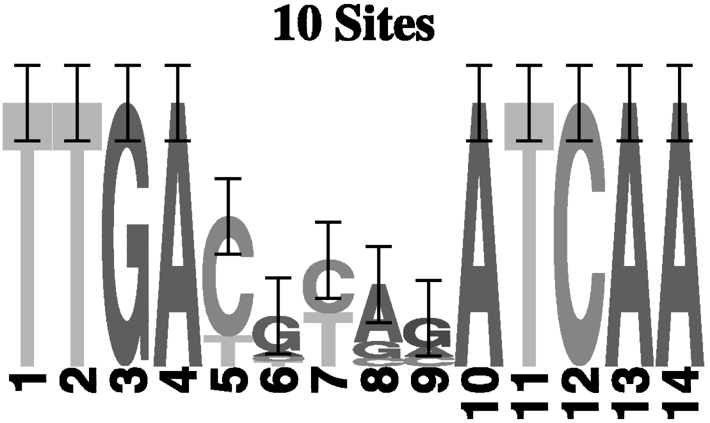

FIG. 6. FNR binding site logo.

parts with respect to the total number of motifs per run in each set.[1] For each of the sets, the consensus sequences, together with the number of similar motifs in the list and the average information content of the best six motifs, are shown. Table 3 clearly shows that the two true motifs are found as the two top-scoring motifs in this test. The other motifs have a low number of occurrences. These numbers deteriorate as shown in Table 4. In this example, we looked for motifs of length 12bp. As was described previously, the scores of the motifs are on average much lower than in the case of motifs of length 8bp. Now, only the consensus of the second motifs is found, but only in 6 out of 100 motifs. If we compare Table 4 with Table 2, we see that 49 motifs are found to overlap with `TCACGTG`, but this motif is not found as one of the top-scoring motifs, since a large part of the motif of length 12bp contains noninformative positions and has a low information content.

5.3. Bacterial sequences

Since no precompiled background model was available, a background model of order 1 was compiled from the sequence data. The order of the background model is limited by the number of nucleotides in the data set. We tested several parameters settings. Varying the motif length from 5bp to 14bp had a profound impact on the motif detection. When searching for a short motif, 5–6bp, the Motif Sampler found both consecutive parts, but when the length was increased only one of the two parts was found due to the masking of the site after an iteration. When the motif length was set to 14bp, the Motif Sampler retrieved the consensus sequence of the described motif. The motif logo is shown in Fig. 6. Table 5 gives a more descriptive overview of the results when we searched for a motif of length 14bp that can have 1 copy. The first two columns identify the genes by their accession number and gene name. The third column gives the position of the retrieved site in the input sequence. The fourth column gives the site. In the next column, there is the probability of finding this motif in the sequence. It is clear that all the motifs are found with a high probability score. This means that the Motif Sampler is very confident on finding the motif in the sequences.

The last column in Table 5 indicates whether the site found by the Motif Sampler corresponds to the site described in the annotation of the corresponding sequences. There are six sequences for which the annotated site matches exactly the site retrieved by the Motif Sampler. In one case, the sequence upstream of the gene ccoN, the Motif Sampler retrieves a site, `TTGACGCAGATCAA`, that matches the consensus but that is located at another position than the one in the GenBank annotation. The site found in the annotation, `AGTTTCACCTGCATC`, differs strongly from the consensus. In three other sequences, there is no element annotated in the GenBank entry, but the Motif Sampler finds a motif occurrence.

[1] A smaller set is contained in the larger sets. The set of 200 motifs contains the set of the first 100 motifs together with the set of 100 motifs found as the second motif in an a given test.

Table 5. Detection of the FNR O_2-Responsive Element

Accn	*Gene*	*Position*	*Site*	*Prob.*	*Annotation*
af016223	ccoN	60	TTGACGCGGATCAA	1.0000	—
af054871	cytN	255	TTGACGTAGATCAA	1.0000	match
pdu34353	ccoN	131	TTGACGCAGATCAA	1.0000	?
pdu34353	ccoG	210	TTGACGCAGATCAA	1.0000	match
af195122	bchE	82	TTGACATGCATCAA	0.9998	—
af016236	dorS	8	TTGACGTCAATCAA	1.0000	—
ae000220	narK	267	TTGATTTACATCAA	0.9986	match
z80340	fixNc	104	TTGATGTAGATCAA	1.0000	match
z80339	fixNd	240	TTGACGCAGATCAA	1.0000	match
pdu34353	fnr	36	TTGACCCAAATCAA	0.9999	match

5.4. *Plant wounding data*

The previous examples can be seen as proof-of-concept tests to illustrate the performance of the Motif Sampler. To analyze a real life problem, the Motif Sampler was run on four sets of upstream regions from coexpressed genes. In these tests, we looked for six different motifs of length 8 and 12bp which can have 1 copy. To distinguish stable motifs from motifs found by chance, we repeated each experiment 10 times and only those motifs were selected that occur in at least five runs. The results with the *A. thaliana* background model of order 3 are shown in Table 6, since they gave the most promising results. We repeated the same tests with the single nucleotide background model, but these results were not as promising as the ones shown in Table 6. In the first column, we give the cluster number and the number of sequences in the cluster. The second column shows the consensus sequences, which are a compilation of the consensus sequences of length 8 and 12bp. Only the relevant part of the consensus is displayed. Together with the consensus, the number of times the consensus was found in 10 runs is indicated. The most frequent motifs are shown here.

Table 6. Results of the Motif Search in 4 Clusters for the Third Order Background Model[a]

Cluster	*Consensus*	*Runs*	*PlantCARE*	*Descriptor*
1 (11 seq.)	TAArTAAGTCAC	7/10	TGAGTCA	tissue specific GCN4-motif
			CGTCA	MeJA-responsive element
	ATTCAAATTT	8/10	ATACAAAT	element associated to GCN4-motif
	CTTCTTCGATCT	5/10	TTCGACC	elicitor responsive element
2 (6 seq.)	TTGACyCGy	5/10	TGACG	MeJa responsive element
			(T)TGAC(C)	Box-W1, elicitor responsive element
	mACGTCACCT	7/10	CGTCA	MeJA responsive element
			ACGT	Abscisic acid response element
3 (5 seq.)	wATATATATmTT	5/10	TATATA	TATA-box like element
	TCTwCnTC	9/10	TCTCCCT	TCCC-motif, light response element
	ATAAATAkGCnT	7/10	—	—
4 (5 seq.)	yTGACCGTCCsA	9/10	CCGTCC	meristem specific activation of H4 gene
			CCGTCC	A-box, light or elicitor responsive element
			TGACG	MeJA responsive element
			CGTCA	MeJA responsive element
	CACGTGG	5/10	CACGTG	G-box light responsive element
			ACGT	Abscisic acid response element
	GCCTymTT	8/10	—	—
	AGAATCAAT	6/10	—	—

[a] In the second column, the consensus of the found motif is given together with the number of times this motif was found in the 10 runs. Finally, the corresponding motif in PlantCARE and a short explanation of the described motif is given.

To assign a potential functional interpretation to the motifs, the consensus of the motifs was compared with the entries described in PlantCARE (Lescot *et al.*, 2002). We found several interesting motifs: methyl jasmonate (MeJa) responsive elements, elicitor-responsive elements, and the abscisic acid response element (ABRE). It is not surprising to find these elements in gene promoters induced by wounding, because there is a clear cross-talk between the different signal pathways leading to the expression of inducible defence genes (Birkenmeier and Ryan, 1998; Reymond and Farmer, 1998; Rouster *et al.*, 1997; Pena-Cortes *et al.*, 1989). Depending on the nature of a particular aggressor (wounding/insects, fungi, bacteria, virus) the plant fine-tunes the induction of defence genes either by employing a single signal molecule or by a combination of the three regulators jasmonic acid, ethylene, and salicylic acid. In the third and fourth cluster, there are also some strong motifs found that do not have a corresponding motif in PlantCARE. These motifs look promising but need some further investigation.

6. CONCLUSION

We have introduced a modified version of the original Gibbs Sampler algorithm to detect regulatory elements in the upstream region of DNA sequences, and we have presented two specific modifications. The first change is the use of a probability distribution to estimate the number of copies of a motif in each sequence. The second contribution is the inclusion of a higher-order background model instead of using single nucleotide frequencies as the background model. These two modifications are incorporated in the iterative scheme of the algorithm that is presented in this paper.

In this manuscript, we showed that our implementation of the Gibbs sampling algorithm for motif finding is able to find overrepresented motifs in several well-described test sets of upstream sequences. We focused on the influence of the different parameters on the performance of the algorithm. We explored different methodologies to analyze the motifs retrieved with our motif-finding algorithm. We tested our implementation on three sets of upstream sequences in which one or more known regulatory elements are present. These data sets allowed us to quantify, up to a certain level of confidence, the performance of our Motif Sampler. The tests showed that the performance increases if the parameters better match the true motif occurrence. Finally, to test the biological relevance of our algorithm, we also used our Motif Sampler to find motifs in sets of coexpressed genes from a microarray experiment. Here we selected those motifs that regularly occur in the repeated tests done on the sets of upstream sequences of each selected cluster. By comparing the selected motifs with the known motifs in PlantCARE, we could identify some interesting motifs among those selected motifs, which are involved in plant defense and stress response.

The algorithm is accessible through a web interface where only a limited number of parameters has to be set by the user. These parameters are simply defined and easy to interpret. Users do not need to go through the details of the implementation to understand how to choose reasonable parameter settings.

Furthermore, we will extend the implementation to improve the usability and performance. First, we will implement a method to automatically detect the optimal length of the motif. Currently, the length of the motif is defined by the user and kept fixed during sampling. Second, we will further optimize the procedure to find the number of copies of the motif in the sequence, which is closely related to the improvement of the motif scores. The ultimate goal is to develop a robust algorithm with as few user-defined parameters as possible. When the parameters are handled well, we will focus on the development of more-specific motif models, such as short palindromic motifs separated by a small variable gap. Also, the combined occurrence of motifs is of greater importance. In the current setup of the algorithm, we will find only individual motifs in consecutive iterations, and there is no clear connection between these motifs. From a biological point of view, it will be very interesting to find significant combinations of motifs in those sets of coexpressed genes. Moreover, the probabilistic framework in which the Motif Sampler is implemented is well suited to incorporate prior biological knowledge in the sequence model (such as when we already know a few examples of the motif but there is not yet enough information to build a complete model).

ACKNOWLEDGMENTS

Gert Thijs is research assistant with the IWT (Institute for the Promotion of Innovation by Science and Technology in Flanders); Kathleen Marchal and Yves Moreau are postdoctoral researchers of the FWO

(Fund for Scientific Research—Flanders); Prof. Bart De Moor is full time professor at the K.U.Leuven. Pierre Rouzé is Research Director of INRA (Institut National de la Recherche Agronomique, France). This work is partially supported by IWT projects STWW-980396; Research Council K.U.Leuven: GOA Mefisto-666; FWO project: G.0115.01; IUAP P4-02 (1997-2001). The scientific responsibility is assumed by its authors. We like to thank the reviewers of this manuscript for their valuable comments.

REFERENCES

Altman, R.B., and Raychaudhuri, S. 2001. Whole-genome expression analysis: Challenges beyond clustering. *Curr. Opin. Struct. Biol.* 11, 340–347.

Audic, S., and Claverie, J-M. 1997. Detection of eukaryotic promoters using Markov transition matrices. *Comput. Chem.* 21(4), 223–227.

Bailey, T. L., and Elkan, C. 1995. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning* 21, 51–80.

Birkenmeier, G.F., and Ryan, C.A. 1998. Wound signaling in tomato plants. Evidence that aba is not a primary signal for defense gene activation. *Plant Physiol.* 117(2), 687–693.

Bucher, P. 1999. Regulatory elements and expression profiles. *Curr. Opin. Struct. Biol.* 9, 400–407.

De Smet, F., Marchal, K., Mathys, J., Thijs, G., De Moor, B., and Moreau, Y. 2002. Adaptive Quality-Based Clustering of Gene Expression Profiles. *Bioinformatics*, in press.

Delcher, A.L., Harman, D., Kasif, S., White, O., and Salzberg, S.L. 1999. Improved microbial gene identification with glimmer. *Nucl. Acids Res.* 27(23), 4636–4641.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95, 14863–14868.

Heyer, L.J., Kruglyak, S., and Yooseph, S. 1999. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Res.* 9, 1106–1115.

Hughes, J.D., Estep, P.W., Tavazoie, S., and Church, G.M. 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.* 296, 1205–1214.

Jensen, L.J., and Knudsen, S. 2000. Automatic discovery of regulatory patterns in promoter regions based on whole cell expression data and functional annotation. *Bioinformatics* 16(4), 326–333.

Krogh, A. 1997. Two methods for improving performance of an HMM and their application for gene finding. *5th Intl. Conf. Intelligent Systems in Molecular Biology*, 179–186.

Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., and Wootton, J.C. 1993. Detecting subbtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.

Lescot, M., Déhais, P., Thijs, G., Marchal, K., Moreau, Y., de Peer, Y. Van, Rouzé, P., and Rombauts, S. 2002. PlantCARE, a database of plant *cis*-acting regulatory elements and a portal to tools for in silico analysis of promoter sequences. *Nucl. Acids Res.* 30, 325–327.

Liu, J.S., Neuwald, A.F., and Lawrence, C.E. 1995. Bayesian models for multiple local sequence alignment and gibbs sampling strategies. *J. Am. Stat. Assoc.* 90(432), 1156–1170.

Liu, X., Brutlag, D.L., and Liu, J.S. 2001. Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Proc. Pacific Symposium on Biocomputing*, vol. 6, 127–138.

Lockhart, D.J., and Winzeler, E.A. 2000. Genomics, gene expression and DNA arrays. *Nature* 405, 827–836.

Lukashin, A.V., and Borodowsky, M. 1998. GeneMark.hmm: New solutions for gene finding. *Nucl. Acids Res.* 26, 1107–1115.

Marchal, Kathleen. 1999. *The O_2 paradox of Azospirillum brasilense under diazotrophic conditions*. PhD. Thesis, FLTBW, KULeuven.

McCue, L.A., Thompson, W., Carmack, C.S., Ryan, M.P., Liu, J.S., Derbyshire, V., and Lawrence, C.E. 2001. Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucl. Acids Res.* 29, 774–782.

Mjolsness, E., Mann, T., Castaño, R., and Wold, B. 2000. From coexpression to coregulation: An approach to inferring transcriptional regulation among gene classes from large-scale expression data. S.A. Solla, T.K. Leen, and K.R. Muller, eds. *Advances in Neural Information Processing Systems*, vol. 12.

Ohler, U., Harbeck, S., Niemann, H., Nöth, E., and Reese, M.G. 1999. Interpolated Markov chains for eukaryotic promotor recognition. *Bioinformatics* 15(5), 362–369.

Ohler, U., and Niemann, H. 2001. Identification and analysis of eukaryotic promoters: Recent computational approaches. *Trends Genet.* 17(2), 56–60.

Pavy, N., Rombauts, S., Déhais, P., Mathé, C., Ramana, D.V.V., Leroy, P., and Rouzé, P. 1999. Evaluation of gene prediction software using a genomic data set: Application to *Arabidopsis thaliana* sequences. *Bioinformatics* 15, 887–899.

Pena-Cortes, H., Sanchez-Serrano, J.J., Mertens, R., Willmitzer, L., and Prat, S. 1989. Abscisic acid is involved in the wound-induced expression of the proteinase inhibitor II gene potato and tomato. *Proc. Natl. Acad. Sci USA* 86, 9851–9855.

Reymond, P., and Farmer, E.E. 1998. Jasmonate and salicylate as global signals for defense gene expression. *Curr. Opin. Plant Biol.* 1(5), 404–411.

Reymond, P., Weber, H., Damond, M., and Farmer, E.E. 2000. Differential gene expression in response to mechanical wounding and insect feeding in Arabidopsis. *Plant Cell* 12, 707–719.

Roth, F.P., Hughes, J.D., Estep, P.W., and Church, G.M. 1998. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole genome mRNA quantitation. *Nature Biotech.* 16, 939–945.

Rouster, J., Leah, R., Mundy, J., and Cameron-Mills, V. 1997. Identification of a methyl jasmonate-responsive region in the promoter of a lipoxygenase-1 gene expressed in barley grain. *Plant J.* 11(3), 513–523.

Sherlock, G. 2000. Analysis of large-scale gene expression data. *Curr. Opin. Immunol.* 12, 201–205.

Sinha, S., and Tompa, M. 2000. A statistical method for finding transcription factor binding sites. *8th Intl. Conf. Intelligent Systems for Molecular Biology*, vol. 8, 37–45.

Szallasi, Z. 1999. Genetic network analysis in light of massively parallel biological data acquisition. *Proc. Pacific Symposium on Biocomputing 99*, vol. 4, 5–16.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., and Church, G.M. 1999. Systematic determination of genetic network architecture. *Nature Genet.* 22(7), 281–285.

Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouzé, P., and Moreau, Y. 2001. A higher order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* 17(12), 1113–1122.

Thijs, G., Moreau, Y., Smet, F. De, Mathys, J., Lescot, M., Rombauts, S., Rouzé, P., De Moor, B., and Marchal, K. 2002. INCLUSive: INtegrated CLustering, Upstream sequence retrieval and motif Sampling. *Bioinformatics* 18(2), 331–332.

Tompa, M. 1999 (August). An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *7th Intl. Conf. Intelligent Systems for Molecular Biology*, vol. 7, 262–271.

van Helden, J., André, B., and Collado-Vides, L. 1998. Extracting regulatory sites from upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* 281, 827–842.

van Helden, J., Rios, A.F., and Collado-Vides, J. 2000a. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.* 28(8), 1808–1818.

van Helden, J., André, B., and Collado-Vides, L. 2000b. A web site for the computational analysis of yeast regulatory sequences. *Yeast* 16, 177–187.

Vanet, A., Marsan, L., Labigne, A., and Sagot, M.F. 2000. Inferring regulatory elements from a whole genome. An analysis of Helicobacter pylori $sigma^{80}$ family of promoter signals. *J. Mol. Biol.* 297(2), 335–353.

Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., and Somogyi, R. 1998. Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA* 95, 334–339.

Workman, C.T., and Stormo, G.D. 2000. ANN-SPEC: A method for discovering transcription binding sites with improved specificity. *Proc. Pacific Symposium on Biocomputing*, vol. 5, 464–475.

Zhang, M.Q. 1999. Large-scale gene expression data analysis: A new challenge to computational biologists. *Genome Res.* 9, 681–688.

Zhu, J., and Zhang, M.Q. 2000. Cluster, function and promoter: analysis of yeast expression array. *Proc. Pacific Symposium on Biocomputing*, vol. 5, 467-486.

Address correspondence to:
Gert Thijs
K.U. Leuven
ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven, Belgium

E-mail: GertThijs@esat.kuleuven.ac.be