BIOINFORMATICS

Vol. 19 no. 5 2003, pages 607–617
DOI: 10.1093/bioinformatics/btg037

Greedy mixture learning for multiple motif discovery in biological sequences

Konstantinos Blekas*, Dimitrios I. Fotiadis and Aristidis Likas

Department of Computer Science, University of Ioannina 45110 Ioannina, Greece and Biomedical Research Institute, Foundation for Research and Technology, Hellas, 45110 Ioannina, Greece

Received on January 24, 2002; revised on April 20, 2002; June 20, 2002; accepted on October 7, 2002

ABSTRACT

Motivation: This paper studies the problem of discovering subsequences, known as *motifs*, that are common to a given collection of related biosequences, by proposing a greedy algorithm for learning a mixture of motifs model through likelihood maximization. The approach adds sequentially a new motif to a mixture model by performing a combined scheme of global and local search for appropriately initializing its parameters. In addition, a hierarchical partitioning scheme based on kd-trees is presented for partitioning the input dataset in order to speed-up the global searching procedure. The proposed method compares favorably over the well-known MEME approach and treats successfully several drawbacks of MEME.

Results: Experimental results indicate that the algorithm is advantageous in identifying larger groups of motifs characteristic of biological families with significant conservation. In addition, it offers better diagnostic capabilities by building more powerful statistical motif-models with improved classification accuracy.

Availability: Source code in Matlab is available at http://www.cs.uoi.gr/~kblekas/greedy/GreedyEM.html

Contact: kblekas@cs.uoi.gr;

1 INTRODUCTION

In protein sequence analysis motif identification is one of the most important problems covering many application areas. It is related to the discovery of portions of protein strands of major biological interest with important structural and functional features. For example, conserved blocks within groups of related sequences (*families*) can often highlight features which are responsible for structural similarity between proteins and can be used to predict the three dimensional structure of a protein. Motifs may also enclose powerful diagnostic features, generating rules for determining whether or not an unknown sequence belongs to a family and thus define a characteristic function for families.

Patterns or motifs are usually distinguished into two general classes: deterministic and probabilistic (Brazma *et al.*, 1998; Brejova *et al.*, 2000). A *deterministic* motif encloses grammatical inference properties in order to describe syntactically a conserved region of related sequences using an appropriate scoring function based on matching criteria. Special symbols, such as arbitrary characters, wild-cards and gaps of variable length can be further used to extend the expressive power of deterministic patterns allowing a certain number of mismatches. The PROSITE database (Hofmann *et al.*, 1999) consists of a large collection of such patterns used to identify protein families. A *probabilistic* motif is described by a model that assigns a probability to the match between the motif and a sequence. The *position weight matrix* (PWM) provides a simplified model of probabilistic *ungapped* motifs representing the relative frequency of each character at each motif position. The ungapped mode suggests that the motif contains a sequence of statistically significant characters (*contiguous motif*) and corresponds to local regions of biological interest. Examples of more complicated probabilistic motifs (allowing gaps, insertions and/or deletions) are profiles and Hidden Markov models (Durbin *et al.*, 1998).

Many computational approaches have been introduced for the problem of motif identification in a set of biological sequences which are classified according to the type of motifs discovered. Excellent surveys (Brazma *et al.*, 1998; Rigoutsos *et al.*, 2000; Brejova *et al.*, 2000) in the literature cover several motif discovery techniques. The Gibbs sampling (Lawrence *et al.*, 1993), MEME (Bailey and Elkan, 1995), TEIRESIAS (Rigoutsos and Floratos, 1998), SAM (Hughey and Krogh, 1998), SPLASH (Califano, 2000) and probabilistic suffix trees (Berejano and Yona, 2001) represent methods for finding multiple shared motifs within a set of unaligned biological sequences.

Among those, the MEME algorithm fits a two-component finite mixture model to a set of sequences using the *Expectation Maximization* (EM) algorithm

*To whom correspondence should be addressed.

(Dempster *et al.*, 1977), where one component describes the motif (ungapped substrings) and the other describes the background (other positions in the sequences). Multiple motifs are discovered by sequentially applying a new mixture model with two components to the sequences remaining after erasing the occurrences of the already identified motifs. Therefore the MEME approach does not allow the parameters of this motif to be reestimated in future steps, and thus future discovered motifs cannot contribute to possible re-allocation of the letter distribution in the motif positions. This drawback becomes significant in the case where there exist motifs that partially match, since these motifs are recognized by the MEME algorithm as one ‘composite’ motif that cannot be further analyzed due to the removal of the motif occurrences.

This paper presents an innovative approach for discovering significant motifs in a set of sequences based on recently developed incremental schemes for Gaussian mixture learning (Li and Barron, 2000; Vlassis and Likas, 2002). Our method learns a mixture of motifs model in a greedy fashion by incrementally adding components (motifs) to the mixture until reaching some stopping criteria or a desired number of motifs. Starting with one component which models the background, at each step a new component is added corresponding to a candidate motif. The algorithm identifies a good initialization for the parameters of the new motif, by performing *global* search over the input substrings together with *local* search based on partial EM steps for fine tuning of the parameters of the new component. In addition, a hierarchical clustering procedure is proposed based on *k*d-tree techniques (Bentley, 1975; Verbeek *et al.*, 2001) for partitioning the input dataset of substrings. This reduces the time complexity for global searching and therefore accelerates the initialization procedure.

In analogy to the MEME approach, our technique discovers motifs when neither the number of motifs nor the number of occurrences of each motif in each sequence is known. However, the main difference with MEME technique is in the way that the mixture models are applied. Although both methods treat the multiple motif identification problem through mixture learning using the EM algorithm, our approach is able to effectively fit multiple-component mixture models. This is achieved through a combined scheme of global and local search, which overcomes the problem of poor initialization of EM that frequently gets stuck on local maxima of the likelihood function. Therefore efficient exploration of the dataset is attained and larger groups of motifs are discovered.

In Section 2 the proposed greedy mixture learning approach for motif discovery in a set of sequences is presented. In addition a novel technique for partitioning the data space in order to reduce the time complexity of global searching is included. Section 3 presents experimental results considering both artificial and real biological datasets. The comparative results with the MEME algorithm indicate the superiority of the proposed Greedy EM and establish its ability to generate more powerful diagnostic signatures. Finally, Section 4 summarizes the proposed method and addresses directions for future research.

2 GREEDY EM ALGORITHM FOR MOTIF DISCOVERY

2.1 The mixture of motifs model

Consider a finite set of characters $\Sigma = \{\alpha_1, \ldots, \alpha_\Omega\}$ where $\Omega = |\Sigma|$. Any sequence $S = a_1 a_2 \ldots a_L$, such that $L \geq 1$ and $a_i \in \Sigma$, is called a *string* (or *sequence*) over the character set Σ, from position 1 to position $L = |S|$. The sequence of characters $a_i a_{i+1} \ldots a_{i+W-1}$ form a *substring* x_i of S length W, identified by the starting position i over the string S. There are $n = L - W + 1$ such possible substrings of length W generated from sequence S.

The probability of the sequence S is[†]:

$$P(S) = P(a_{W-1} \ldots a_1) P(a_W | a_{W-1} \ldots a_1) \ldots \\ P(a_L | a_{L-1} \ldots a_1), \qquad (1)$$

which can be approximately written as

$$P(S) \approx P(a_{W-1} \ldots a_1) P(a_W | a_{W-1} \ldots a_1) \ldots \\ \ldots P(a_L | a_{L-1} \ldots a_{L-W+1}) \\ = \frac{P(a_W \ldots a_1) P(a_{W+1} \ldots a_2) \ldots P(a_L \ldots a_{L-W+1})}{P(a_W \ldots a_2) P(a_{W+1} \ldots a_3) \ldots P(a_{L-1} \ldots a_{L-W+1})} . \qquad (2)$$

By defining $y_i = a_i a_{i+1} \ldots a_{i+W-2}$ (substrings of length $W - 1$) we obtain the following log likelihood $\mathcal{L}$:

$$\mathcal{L} = \log P(S) = \sum_{i=1}^{L-W+1} \log P(x_i) - \sum_{i=2}^{L-W+1} \log P(y_i) . \qquad (3)$$

The first term, which has to be maximized, is the log likelihood of a set of independent substrings of length W. This can be accomplished with the EM algorithm, as shown below. The second term, which has to be minimised, is the log likelihood of a set of independent substrings of length $W - 1$. This contribution corrects for the fact that our substrings are strongly overlapping and therefore *not* dependent. Since the minimization of the log likelihood would have to resort to a slow iterative procedure, we adopt the following approximate method. First, the dependence of the substrings is neglected, that

[†] Equations (1)–(3) were suggested by the anonymous referees.

is, the second term is discarded and $\mathcal{L}$ is approximated by

$$\mathcal{L} \approx \sum_{i=1}^{L-W+1} \log P(x_i). \tag{4}$$

The effect of the second, neglected, term is then heuristically incorporated into the optimization procedure by discarding substrings that show a significant overlap with other motifs. This is discussed in Section 2.3.

We assume a set of N unaligned sequences $\boldsymbol{S} = \{S_1, \ldots S_N\}$ of length $L_1, \ldots, L_N$, respectively. In order to deal with the problem of identifying motifs of length W we construct a new dataset containing all substrings of length W in $\boldsymbol{S}$. Since for each original sequence S_s (of length L_s) there are $m_s = L_s - W + 1$ possible substrings of length W, we obtain a training dataset $X = \{x_1, \ldots, x_n\}$ of n substrings ($n = \sum_{s=1}^{N} m_s$) for the learning problem.

A mixture of motifs model f for an arbitrary substring x_i assuming g components can be written as

$$f(x_i; \Psi_g) = \sum_{j=1}^{g} \pi_j \phi_j(x_i; \theta_j), \tag{5}$$

where Ψ_g is the vector of all unknown parameters in the mixture model of g components, i.e. $\Psi_g = [\pi_1, \ldots, \pi_{g-1}, \theta_1, \ldots, \theta_g]$. The mixing proportion π_j ($\pi_j \geq 0$, $\forall j = 1, \ldots, g$) can be viewed as the prior probability that data x_i has been generated by the jth component of the mixture. It holds that $\sum_{j=1}^{g} \pi_j = 1$.

Each one of the g components corresponds to either a motif or the background. A motif j can be modeled by a position weight matrix $\mathrm{PWM}_j = [p_{l,k}^j]$ of size $[\Omega \times W_j]$, where each value $p_{l,k}^j$ denotes the probability that the letter α_l is located in motif position k. Although the general model considers motifs of variable length W_j, in the sequel we assume motifs of constant length W. On the other hand, a background component j is represented using a probability vector BPM_j (of length Ω), where each parameter value ϱ_l^j denotes the probability of letter α_l to occur at an arbitrary position. The probability that a substring $x_i = a_{i1} \ldots a_{iW}$, where $a_{ik} \in \Sigma$ ($k = 1, \ldots, W$) has been generated by the component j is

$$\phi_j(x_i; \theta_j) = \begin{cases} \prod_{k=1}^{W} p_{a_{ik},k}^j & \text{if } j \text{ is motif} \\ \prod_{k=1}^{W} \varrho_{a_{ik}}^j & \text{if } j \text{ is background,} \end{cases} \tag{6}$$

where the probability matrix PWM_j (or BPM_j) corresponds to the parameter vector θ_j.

The log-likelihood of the observed dataset X corresponding to the above model is

$$\mathcal{L}(\Psi_g) = \sum_{i=1}^{n} \log f(x_i; \Psi_g). \tag{7}$$

Formulating the problem as an incomplete-data problem (McLachlan and Peel, 2001), each substring x_i can be considered as having arisen from one of the g components of the mixture model of Equation (5). We can define the parameters $z_{ij} = 1$ or 0 (missing parameters) indicating whether x_i has been generated by the j-th component of the mixture ($i = 1, \ldots, n$; $j = 1, \ldots, g$). Then, the *complete*-data log-likelihood $\mathcal{L}^c$ is given by

$$\mathcal{L}^c(\Psi_g) = \sum_{i=1}^{n} \sum_{j=1}^{g} z_{ij} \{\log \pi_j + \log \phi_j(x_i; \theta_j)\}. \tag{8}$$

The EM algorithm can be applied for the log-likelihood maximization problem by treating z_{ij} as missing data. The following update equations are obtained for each component j (Render and Walker, 1984; Bailey, 1995; Bailey and Elkan, 1995)

$$z_{ij}^{(t+1)} = Pr(z_{ij} = 1 | x_i, \Psi_g^{(t)}) = \frac{\pi_j^{(t)} \phi_j(x_i; \theta_j^{(t)})}{f(x_i; \Psi_g^{(t)})}, \tag{9}$$

$$\pi_j^{(t+1)} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} z_{ij}^{(t+1)}, \tag{10}$$

$$\theta_j^{(t+1)} = \begin{cases} \hat{p}_{l,k}^j = \dfrac{\hat{c}_{l,k}^j}{\sum_{l=1}^{\Omega} \hat{c}_{l,k}^j} & \text{if } j \text{ is motif} \\ \hat{\varrho}_l^j = \dfrac{\hat{c}_l^j}{\sum_{l=1}^{\Omega} \hat{c}_l^j} & \text{if } j \text{ is background,} \end{cases} \tag{11}$$

where the elements $\hat{c}_{l,k}^j$ ($\hat{c}_l^j$) correspond to the observed frequency of letter α_l at position k of motif j occurrences (at background j arbitrary positions) and can be formally expressed as

$$\begin{aligned} \hat{c}_{l,k}^j &= \sum_{i=1}^{n} z_{ij}^{(t+1)} \boldsymbol{I}(a_{ik}, l) && \text{if } j \text{ is motif} \\ \hat{c}_l^j &= \sum_{i=1}^{n} z_{ij}^{(t+1)} \sum_{k=1}^{W} \boldsymbol{I}(a_{ik}, l) && \text{if } j \text{ is background.} \end{aligned} \tag{12}$$

The indicator $\boldsymbol{I}(a_{ik}, l)$ denotes a binary function which is given as

$$\boldsymbol{I}(a_{ik}, l) = \begin{cases} 1 & \text{if } a_{ik} \equiv \alpha_l \text{ (letter } \alpha_l \text{ is at position } k \text{ of } x_i) \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Equations (9)–(11) can be used to estimate the parameter values Ψ_g of the g-component mixture model which maximize the log-likelihood function [Equation (8)]. After training the mixture model, the motif occurrences (substrings x_i) can be obtained using the estimated posterior probabilities z_{ij} [Equation (9)]. Since it has been shown that the application of the EM algorithm to mixture problems monotonically increases the likelihood function (Dempster *et al.*, 1977), these EM steps ensure the convergence of the algorithm to a local maximum of the likelihood function in most cases (in some rare cases this can also be a saddle point). However, its great dependence on parameter initialization and its local nature (it gets stuck in local maxima of the likelihood function) do not allow us to directly apply the EM algorithm to a g-component mixture of motifs model.

To overcome the problem of poor initialization of the model parameters, several techniques have been introduced (McLachlan and Peel, 2001). The MEME approach, for example, uses a dynamic programming algorithm which estimates the goodness of many possible starting points based on the likelihood measurement of the model after one iteration of EM (Bailey, 1995; Bailey and Elkan, 1995). Our method provides a more efficient combined scheme by applying global search over appropriate defined candidate motifs, followed by a local search for fine tuning the parameters of a new motif. In Section 2.2 we describe a procedure for properly adding a new motif and it is shown how the monotone increase of the likelihood can be guaranteed.

2.2 Greedy mixture learning

Assume that a new component $\phi_{g+1}(x_i;\theta_{g+1})$ is added to a g-component mixture model $f(x_i;\Psi_g)$. The new component corresponds to a motif modeled by the position weight matrix PWM$_{g+1}$ denoted by the parameter vector θ_{g+1}. Then the resulting mixture has the following form:

$$f(x_i;\Psi_{g+1}) = (1-a)f(x_i;\Psi_g) + a\phi_{g+1}(x_i;\theta_{g+1}), \tag{14}$$

with $a \in (0,1)$. Ψ_{g+1} specifies the new parameter vector and consists of the parameter vector Ψ_g of the g-component mixture, the weight a and the parameter vector θ_{g+1}. Then, the log-likelihood for Ψ_{g+1} is given by

$$\mathcal{L}(\Psi_{g+1}) = \sum_{i=1}^{n} \log f(x_i;\Psi_{g+1}) = \sum_{i=1}^{n} \log\{(1-a)f(x_i;\Psi_g) + a\phi_{g+1}(x_i;\theta_{g+1})\}. \tag{15}$$

The above formulation proposes a two-component likelihood maximization problem, where the first component is described by the old mixture $f(x_i;\Psi_g)$ and the second one is the motif component $\phi_{g+1}(x_i;\theta_{g+1})$ with $\theta_{g+1} = [p_{l,k}^{g+1}]$ ($l = 1,\ldots,\Omega$; $k = 1,\ldots,W$) describing the position weight matrix PWM$_{g+1}$. If we consider that the parameters Ψ_g of $f(x_i;\Psi_g)$ remain fixed during maximization of $\mathcal{L}(\Psi_{g+1})$, the problem can be treated by applying searching techniques to optimally specify the parameters a and θ_{g+1} which maximize $\mathcal{L}(\Psi_{g+1})$.

An efficient technique for the specification of θ_{g+1} is presented in Vlassis and Likas (2002) using a combination of local and global searching. In particular, an EM algorithm performs local search for the maxima of likelihood with respect to a and θ_{g+1}, where the learning procedure is applied only to the mixing weight a and the probabilistic quantities $p_{l,k}^{g+1}$ of the newly inserted component (motif-model). Following Equations (9)–(11) and assuming that the new component describes a motif, the following update procedures can be derived.

$$z_{i,g+1}^{(t+1)} = Pr(z_{i,g+1} = 1|x_i, \theta_{g+1}^{(t)}, a^{(t)}) = \frac{a^{(t)}\phi_{g+1}(x_i;\theta_{g+1}^{(t)})}{(1-a^{(t)})f(x_i;\Psi_g)+a^{(t)}\phi_{g+1}(x_i;\theta_{g+1}^{(t)})}, \tag{16}$$

$$a^{(t+1)} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} z_{i,g+1}^{(t+1)}, \tag{17}$$

$$\theta_{g+1}^{(t+1)} = [\hat{p}_{l,k}^{g+1}], \text{ where } \hat{p}_{l,k}^{g+1} = \frac{\hat{c}_{l,k}^{g+1}}{\sum_{l=1}^{\Omega}\hat{c}_{l,k}^{g+1}}, \tag{18}$$

where

$$\hat{c}_{l,k}^{g+1} = \sum_{i=1}^{n} z_{i,g+1}^{(t+1)}\boldsymbol{I}(a_{ik}, l). \tag{19}$$

The above *partial* EM steps constitute a simple and fast method for local searching the maxima of $\mathcal{L}(\Psi_{g+1})$. However, the problem of poor initialization still remains since this scheme is very sensitive to the proper initialization of the two parameters a and θ_{g+1}. For this reason a global search strategy has been developed (Vlassis and Likas, 2002) which facilitates the global search over the parameter space. In particular, by substituting the log-likelihood function [Equation (15)] using a Taylor approximation about a point $a = a_0$, we can search for the optimal θ_{g+1} value from the resulting estimate. Therefore, we expand $\mathcal{L}(\Psi_{g+1})$ by second order Taylor expansion about $a_0 = 0.5$ and then the resulting quadratic function is maximized with respect to a. This results into the following approximation:

$$\hat{\mathcal{L}}(\Psi_{g+1}) = \mathcal{L}(\Psi_{g+1}|a_0) - \frac{[\dot{\mathcal{L}}(\Psi_{g+1}|a_0)]^2}{2\ddot{\mathcal{L}}(\Psi_{g+1}|a_0)}, \tag{20}$$

where $\dot{\mathcal{L}}(\Psi_{g+1})$ and $\ddot{\mathcal{L}}(\Psi_{g+1})$ are the first and second derivatives of $\mathcal{L}(\Psi_{g+1})$ with respect to a. It can be shown (Vlassis and Likas, 2002) that, for a given parameter vector θ_τ, a local maximum of $\mathcal{L}(\Psi_{g+1})$ near $a_0 = 0.5$ is given by

$$\hat{\mathcal{L}}(\theta_\tau) = \sum_{i=1}^{n} \log\frac{f(x_i;\Psi_g) + \phi_{g+1}(x_i;\theta_\tau)}{2} + \frac{1}{2}\frac{[\sum_{i=1}^{n}\delta(x_i,\theta_\tau)]^2}{\sum_{i=1}^{n}\delta^2(x_i,\theta_\tau)}, \tag{21}$$

and is obtained for

$$\hat{a} = \frac{1}{2} - \frac{1}{2}\frac{\sum_{i=1}^{n}\delta(x_i;\theta_\tau)}{\sum_{i=1}^{n}\delta^2(x_i;\theta_\tau)}, \tag{22}$$

where

$$\delta(x_i, \theta_\tau) = \frac{f(x_i; \Psi_g) - \phi_{g+1}(x_i; \theta_\tau)}{f(x_i; \Psi_g) + \phi_{g+1}(x_i; \theta_\tau)}. \tag{23}$$

If the estimated value of a falls outside the range $(0, 1)$ then we can initialize the partial EM with the approximation $\hat{a} = 0.5$ for $g = 1$ and $\hat{a} = 1/(g+1)$ for $g \geq 2$, according to Li and Barron (2000).

The above methodology has the benefit of modifying the problem of maximizing the likelihood function [Equation (15)] to become independent on the selection of initial value for the mixing weight a. In addition, this procedure reduces the parameter search space and restricts global searching on finding good initial values θ_τ of the probability matrix θ_{g+1} (probabilities $p^{g+1}_{l,k}$). The last observation is made clearer from Equation (21) where $\hat{\mathcal{L}}(\theta_\tau)$ depends only on $\phi_{g+1}(x_i; \theta_\tau)$, while $f(x; \Psi_g)$ remains fixed during optimization. The only problem is now the identification of a proper initial value θ_τ so as to conduct partial EM steps. Therefore we need to find *candidates* for the initialization of the motif parameters.

2.3 Candidate selection for initializing new model parameters

A reasonable approach of initializing motif parameters θ_{g+1} is to search for candidates directly over the total dataset of substrings $X = \{x_\tau\}$, $(\tau = 1, \ldots, n)$, where $x_\tau = a_{\tau 1} \ldots a_{\tau W}$. For this reason each substring x_τ is associated with a position weight matrix θ_τ constructed as

$$\theta_\tau = [p^\tau_{l,k}], \text{ where } \; p^\tau_{l,k} = \begin{cases} \lambda & \text{if } a_{\tau k} = \alpha_l \\ \frac{1-\lambda}{\Omega-1} & \text{otherwise.} \end{cases} \tag{24}$$

The parameter λ has a fixed value in the range $(0, 1)$, where its value depends on the Σ alphabet size (Ω) and satisfies $\lambda \geq 1/\Omega$ (e.g. $\lambda \gg 0.05$ for protein sequence alphabet where $\Omega = 20$). Therefore, the (local) log-likelihood $\hat{\mathcal{L}}(\theta_\tau)$ is determined by selecting among the θ_τ-matrices $(\tau = 1, \ldots, n)$ the one which maximizes the right-hand side of Equation (21), i.e.

$$\hat{\theta}_{g+1} = \arg\max_{\theta_\tau} \hat{\mathcal{L}}(\theta_\tau). \tag{25}$$

In order to accelerate the above searching procedure the following quantities $\xi_{\tau,i}$ for each substring $x_i = a_{i1} \ldots a_{iW}$ are computed

$$\xi_{\tau,i} (= \phi_{g+1}(x_i; \theta_\tau)) = \prod_{k=1}^{W} p^\tau_{a_{ik},k}, \tag{26}$$

which substitute for the $\phi_{g+1}(x_i; \theta_\tau)$ in Equations (21) and (23). Following this observation, the searching is made over these quantities $\xi_{\tau,i}$ which maximize Equation (21). The constructed matrix Ξ with elements $\xi_{\tau,i}$ is calculated once during the initialization phase of the learning algorithm and is applied each time a new component is added to a g-component mixture. Similar techniques for searching for global solutions over the parameter space have been proposed in Smola *et al.* (1999); Vlassis and Likas (2002).

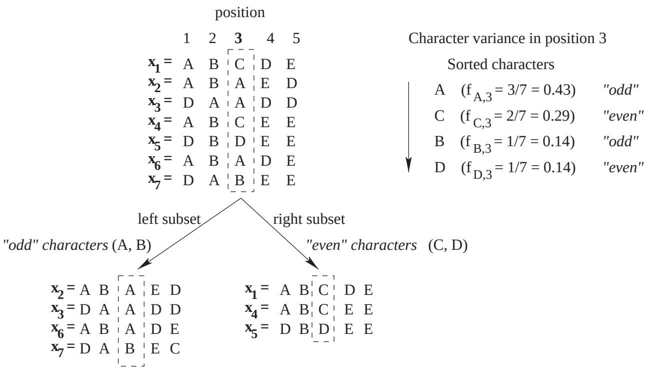

Fig. 1. Partitioning occurs in the *third* position that presents the maximum character variance.

The drawback of searching for candidates over all substrings n of the dataset is the increasing time complexity $(\mathcal{O}(n^2))$ of the search procedure. Indeed, $\mathcal{O}(n^2)$ computations are needed since the likelihood of every substring under every such candidate parameterized model must be computed. In order to reduce the complexity, we perform a *hierarchical clustering* pre-processing phase based on kd-trees. Original kd-trees (Bentley, 1975) were proposed in an attempt to speed-up the execution of nearest neighbor queries by defining a recursive binary partitioning of a k-dimensional dataset, where the root node contains all data. Most such techniques partition the data at each tree level using an appropriate hyperplane perpendicular to the direction which presents major variance of the data (Sproull, 1991; Verbeek *et al.*, 2001).

A modification is proposed in order to deal with sequential data. Starting with a root node that contains the total set of substrings X, at each step we partition the set of substrings at each node using an appropriate criterion. In particular, after calculating the relative frequency values $f_{l,k}$ of each character α_l at each substring position k in the subset of substrings contained in that node, we identify the position q that exhibits the largest entropy value H_k:

$$H_k = - \sum_{\substack{\alpha_l \in \Sigma \\ f_{l,k} > 0}} f_{l,k} \log f_{l,k} \, , \tag{27}$$

and $q = \arg\max_{k=1,\ldots,W}\{H_k\}$. The partitioning procedure is implemented by initially sorting the characters α_l in the position q according to their relative frequency values $f_{l,q}$

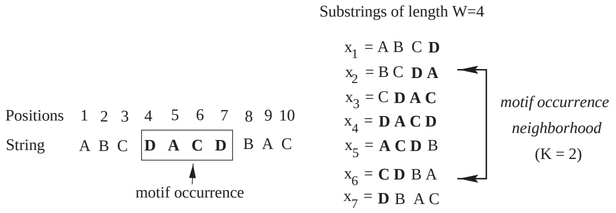

Fig. 2. The neighborhood of a motif occurrence.

and then labeling them as '*odd*' or '*even*'. In this case two subsets are created (left and right), which are successively filled with the substrings that contain the 'odd' (left) and the 'even' (right) characters in the position q. An example is shown in Figure 1 where the third position with the maximum entropy is selected for partitioning.

The above recursive procedure builds a tree with several nodes and the partitioning for a node is terminated (*leaf node*) when the number of included substrings is lower than a fixed value T. Every node of the tree contains a subset (cluster) of the original set of substrings and each such cluster is characterized by its *centroid* (*consensus substring*). Therefore, the total set of leaf nodes consists of C centroids and their corresponding position weight matrices [obtained by Equation (24)] constitute the candidate motif parameters used in global searching. Our experiments have shown that this partitioning technique drastically accelerates the global search procedure without affecting significantly its performance.

Another problem we must face concerns the occurrence of overlappings with the already discovered motifs during the selection of a candidate motif instance. As it has been already discussed in Section 2.1 this is necessary since our objective is to find a set of motifs that do not exhibit *any significant overlap*. Therefore when a new motif is discovered, substrings which correspond to positions next to the motif occurrences in the original set of sequences (determining the *neighborhood* of motif occurrences) contain a portion of the discovered motif. If any of these *overlapping* substrings were used as a candidate motif model for the initialization of a new component, it would probably lead (as the performance of the EM algorithm depends very much on the initialization of the parameters (Dempster *et al.*, 1977; McLachlan and Peel, 2001)) to the discovery of a new motif that would exhibit significant overlap with another one already being discovered. An example is illustrated in Figure 2, where the substrings x_1, ... x_7 of length $W = 4$ overlap with the discovered motif occurrence $x_4 =$ DACD and these substrings should not be subsequently considered as candidates for the discovery of additional motifs.

In order to avoid this inconvenience, a binary indicator value ϖ is introduced for each leaf node ($\tau = 1, \ldots, C$), whose value indicates the occurrence of a significant portion of a motif already being discovered ($\varpi_\tau = 1$) in the subset corresponding to that node. A parameter K ($K < W$) is used to define the neighborhood $\mathcal{N}_i$ of a motif occurrence x_i as the set of substrings x_j ($j = i - K, \ldots, i + K$) which *match* at least K contiguous characters with x_i, and thus are derived from K left and K right starting positions from the starting position i (in the original set of sequences) that corresponds to x_i, i.e.

$$\mathcal{N}_i = \{x_j\},\ j = i - K, \ldots, i + K. \tag{28}$$

For example, setting $K = 2$ the substrings x_j ($j = 2, \ldots, 6$) in Figure 2, are included in the neighborhood of the motif occurrence $x_4 =$ DACD. Initially we set $\varpi_\tau = 0$ ($\forall\tau = 1, \ldots, C$), and whenever a new motif g is found and its motif occurrences x_i are identified[‡], the leaf nodes τ that contain substrings belonging to the neighborhood $\mathcal{N}_i$ of one of the x_i are excluded ($\varpi_\tau = 1$) from the set of candidate motifs used in the global search phase. Best results are obtained for $K < W/2$.

The above strategy eliminates the possibility of overlappings among the motifs discovered and ensures proper specification of candidate motif models, while at the same time it iteratively reduces further the time complexity of the global search. In comparison with the MEME approach where substrings that correspond to the neighborhood of motif occurrences are deleted from the dataset, in our scheme the overlapping substrings are only excluded from the set of candidate motifs used in the global search phase. This constitutes a significant advantage over the MEME approach.

Algorithm 1 summarizes the described ideas for learning a mixture of motifs model for the motif discovery problem. It ensures the monotonic increase of the log-likelihood of the learning set since EM cannot decrease the log-likelihood and the proposed partial EM solutions are accepted only if $\mathcal{L}(\Psi_{g+1}) > \mathcal{L}(\Psi_g)$. The stopping condition of the algorithm depends not only on the maximum allowed number of components g (specified by the user), but also on the vector ϖ. This means that in the case $\varpi_\tau = 1$, $\forall\tau = 1, \ldots, C$, the parameter space for selecting candidate components has been entirely searched and therefore the possibility of existence of another motif in the set of sequences is very low.

3 EXPERIMENTAL RESULTS

The experiments described in this section have been conducted using both artificial[§] and real datasets. In all the experiments the parameter λ [Equation (24)] was set

[‡] The motif occurrences x_i are those with z_{ig} values (Equation (9)) that are close to 1 (e.g. $z_{ig} > 0.9$).

[§] A documentation that describes experiments with artificial datasets can be downloaded at http://www.cs.uoi.gr/∼kblekas/greedy/GreedyEM.html.

Algorithm 1 : The Greedy EM algorithm

```
Start with a set of $N$ unaligned sequences $S = \{S_s\}$ with length $L_s = |S_s|$
($s = 1, \ldots, N$), taking values from alphabet $\Sigma = \{\alpha_1, \ldots, \alpha_\Omega\}$ of length
$\Omega = |\Sigma|$.
  initialize
    • Apply a window of length $W$ to the original set $S$ creating a
      learning dataset of substrings $X = \{x_i\}$ ($i = 1, \ldots, n$), where
      $n = \sum_{s=1}^{N}(L_s - W + 1)$.
    • Apply the proposed $k$d-tree approach over the dataset $X$, find the $C$
      final consensus substrings and define the corresponding $C$ candidate
      initial parameter values $\theta_\tau$, $\tau = 1, \ldots, C$ (Equation 24) used for
      global searching.
    • Initialize $\varpi_\tau = 0$, $\forall\tau = 1, \ldots, C$, and calculate matrix $\Xi$ of
      quantities $\xi_{\tau,i}$ according to Equation 26.
    • Initialize the model using one component ($g = 1$) that represents the
      background with parameter settings (probability matrix $\theta_1$) equal to
      the relative frequencies of characters $\alpha_l \in \Sigma$, i.e. $\varrho_l^1 = \frac{f_l}{\sum_{s=1}^{N} L_s}$,
      where $f_l$ indicates the frequency of character $\alpha_l$ in the set of
      sequences $S$.
  repeat
    1. Perform EM steps (Equations 9-11) until convergence:
       $|\mathcal{L}^{(t)}(\Psi_g)/\mathcal{L}(\Psi_g)^{(t-1)} - 1| < 10^{-6}$.
    2. If $g \geq 2$ then from the motif occurrences $x_i$ (with $z_{ig} > 0.9$)
       find their neighborhood $\mathcal{N}_i = \{x_j\}$ ($j = i - K, \ldots, i + K$) and
       set $\varpi_\tau = 1$ for each leaf node $\tau$ which contains any of the $x_j$.
    3. Insert a new candidate component $g + 1$ by searching over
       all $\theta_\tau$ (where $\tau = 1, \ldots, C$ and $\varpi_\tau = 0$) and setting $\hat{\theta}_{g+1}$
       equal to the $\theta_\tau$ that maximize the log-likelihood function of
       Equation 21 using the already calculated quantities $\xi_{\tau,i}$ instead
       of $\phi(x_i; \theta_\tau)$. Compute the weight $\hat{a}$ using the obtained value
       $\hat{\theta}_{g+1}$ in Equation 22.
    4. Perform partial EM steps (Equations 16 - 18) with initial values
       $\hat{a}$ and $\hat{\theta}_{g+1}$, until convergence as in step 1 and obtain the
       parameter values $\Psi_{g+1}$.
    5. If $\mathcal{L}(\Psi_{g+1}) > \mathcal{L}(\Psi_g)$ then accept the new mixture model with
       $g + 1$ components and go to step 1, otherwise terminate.
  until an appropriate condition is met (e.g. maximum number of motifs, or
  $\sum_{\tau=1}^{C} \varpi_\tau = C$).
```

equal to 0.8, and while we set $K = 1$ to specify the neighborhood of a motif. Finally, the proper value for parameter T (maximum size of the kd-tree leaf nodes) was empirically estimated $T = N/2$ (half the number of sequences).

For all the experimental datasets we have also applied the MEME approach using the available software from the corresponding Web site[¶], where we have selected the ‘any number of repetitions’ model (Bailey, 1995; Bailey and Elkan, 1995) as it is equivalent to the one used in our approach.

3.1 Experiments with real datasets

Our experimental study with real datasets has two objectives. First to measure the effectiveness of our method in discovering already known motifs in real protein families, as well as to explore the possibility of obtaining additional (currently unknown) motifs and therefore build larger groups of characteristic motifs for real families. Second to study the diagnostic capabilities of groups of motifs constructed by the proposed Greedy EM approach, by measuring their classification accuracy within the SWISS-PROT database of protein sequences (Bairoch and Apweiler, 2000). In this spirit, we have selected real datasets from the PRINTS database (Attwood *et al.*, 2000) and the PROSITE database of protein families (Hofmann *et al.*, 1999).

Table 1. The PRINTS datasets selected for our experimental study and their fingerprints as reported in the PRINTS Web site

PRINTS family	Sequences (average length)	PRINTS fingerprints (number and length of motifs)
PR00058	16 (297)	6 ($W = [20, 21]$)
PR00061	24 (120)	4 ($W = [24, 30]$)
PR00810	6 (286)	2 ($W = [10, 11]$)
PR01266	24 (222)	3 ($W = [15, 17]$)
PR01267	22 (218)	4 ($W = [12, 14]$)
PR01268	19 (209)	3 ($W = [17, 22]$)

To evaluate the quality of the discovered motifs we computed the *information content* (IC) (also called relative entropy) (Bailey and Elkan, 1995; Hertz and Stormo, 1999) as follows:

$$IC_j = \sum_{k=1}^{W} \sum_{\alpha_l \in \Sigma} p_{lk}^{j} \log_2 \frac{p_{lk}^{j}}{\bar{f}_{\alpha_l}}, \tag{29}$$

where $\bar{f}_{\alpha_l}$ indicates the overall relative frequency of letter α_l in the training set of sequences. As stated in Hertz and Stormo (1999); Tompa (1999) the IC provides a good measure for comparing motifs having nearly the same number of occurrences, but not in cases where the numbers of motif occurrences are quite different.

3.1.1 Discovering fingerprints for PRINTS datasets The PRINTS database contains protein family *fingerprints* which are groups of motifs that occur *in every family member* and thus are characteristic of a family. The identification of the fingerprints within the PRINTS database has been made using database scanning algorithms from sequence analysis tools Attwood *et al.* (2000). Current release of PRINTS (32.0) contains 1600 families with 9800 individual motifs.

Six (6) PRINTS families have been selected as shown in Table 1. For each family, we conducted experiments for several values of the motif length W. In each experiment we applied both MEME and Greedy EM until at most 15 motifs had been discovered. The final fingerprints provided by each technique included only the motifs that

[¶] The Web site of MEME/MAST system version 3.0 can be found at http://meme.sdsc.edu/meme/website/

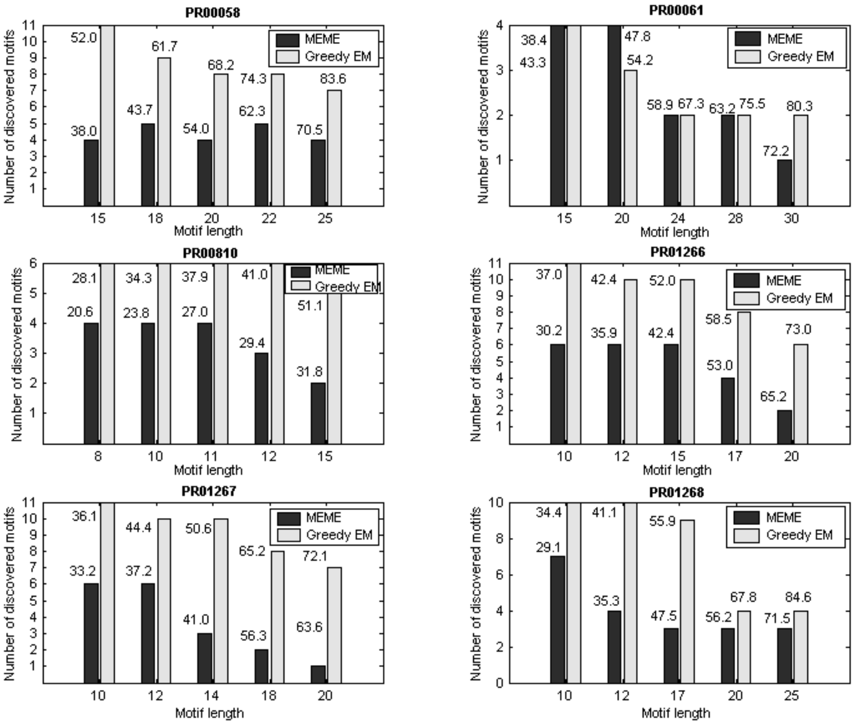

Fig. 3. The number of the motifs discovered by MEME and Greedy EM for several values of the motif length W in each PRINTS dataset. The numbers inside the graph indicate average IC-scores for the discovered fingerprints.

occur once in every family member (single copy per sequence). The remaining motifs were removed from the final set.

Figure 3 displays the obtained results where the vertical bars represent the fingerprint size (number of motifs occurring in every training sequence). Since all motifs exhibit the same number of occurrences in the training sequences, the IC-score can be employed for motif evaluation. For this reason, the average IC-scores of the identified fingerprints are also presented in Figure 3 for several values of the motif length W[‖]. From the results it is clear that the proposed method has the ability to build greater and better conserved fingerprints for almost every value of W.

3.1.2 Measuring the classification accuracy in protein families It has been previously mentioned that one significant reason for identifying characteristic motif-models of sequence families is related to the subsequent employment of the motifs for classification purposes. The MAST algorithm (Bailey and Gribskov, 1998) constitutes an example of a sequence homology searching tool that matches multiple motif models against a set of sequences using as test statistic the product of the *p-values* of motif match score.

The purpose of this series of experiments is to evaluate the quality of motifs in protein families discovered by both the MEME and the Greedy EM approach by measuring their classification accuracy in sets of target sequences. Since the previous experimental results have demonstrated the potential of our approach to discover a larger number of more conservative motifs, we next proceed in using the discovered motifs for sequence homology searching.

We have selected to experiment with four datasets from the PROSITE database of protein sequence families (Hofmann *et al.*, 1999) summarized in Table 2. The sequences of each dataset were considered as positive data, and a small percentage of sequences from each family was randomly selected as the training datasets for motif discovery. A fixed value $W = 10$ has been selected as the length of the motifs and redundant motifs were removed[**] from the set of motifs provided by each algorithm. The numbers of the characteristic motifs discovered for the four PROSITE families are presented in Table 3 where it is again obvious the capability of the Greedy EM approach to identify larger motif groups.

[‖] More details on the discovered PRINTS fingerprints can be found at http://www.cs.uoi.gr/∼kblekas/greedy/GreedyEM.html.

[**] As indicated by the experiments, the redundant motifs have no effect on the performance of the MAST homology detection algorithm.

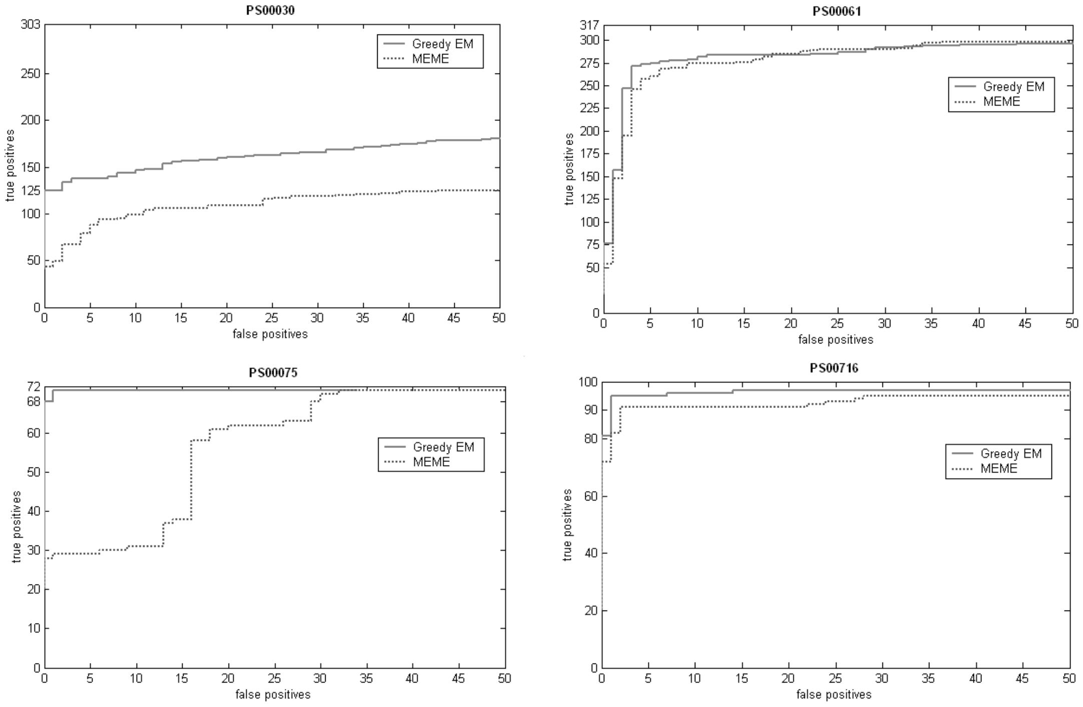

Fig. 4. Receiver operating characteristic (ROC$_{50}$) curves for four different PROSITE families using the Greedy EM (solid line) and the MEME (dot line) for motif discovery and then applying each group of motifs to the MAST homology finding algorithm.

Table 2. The PROSITE families used for the experimental study of the diagnostic significance of the group of the discovered motifs. The entire SWISS-PROT database (release 40, 103990 sequences) was used as the set of target sequences for evaluating the classification performance of the MEME–MAST and the Greedy EM–MAST approaches

SWISS-PROT (test set)	PROSITE family	Positive data	Training set (average length of seqs)
	PS00030	303	15 (467)
103990	PS00061	317	20 (272)
sequences	PS00075	72	14 (218)
	PS00716	100	20 (396)

The MAST homology search algorithm is used to calculate the statistical significance (E-value) of the matches of a group of motifs (characteristic of a protein family) to target real sequences. The experimental methodology (adopted from Bailey and Gribskov (1998)) is the following: After training a set of sequences with MEME and Greedy EM, the MAST software takes as input the set of discovered motifs and computes the E-value for each sequence in the SWISS-PROT database (SWISS-PROT release 40, number of entries 103990 sequences). By specifying a threshold for the E-value we classify as positives those target sequences with E-value lower than the threshold.

Table 3. Number of motifs of length $W = 10$ discovered using the MEME and the Greedy EM algorithm in the four PROSITE families

PROSITE	Number of discovered motifs ($W = 10$)	
family	MEME	Greedy EM
PS00030	4	7
PS00061	5	10
PS00075	7	10
PS00716	13	17

For each experiment we measured the number of false positives and true positives observed at a given E-value threshold in order to estimate the sensitivity and specificity of each method. Figure 4 summarizes the performance of the two techniques in the four real datasets by presenting the receiver operating characteristic curves, i.e. plots of the true positives as a function of false positives for varying E-value thresholds until 50 false positives are found (ROC$_{50}$) (Gribskov and Robinson, 1996). The superior classification performance of the MAST algorithm using the motifs provided by the Greedy EM algorithm is obvious from the plotted curves. In PROSITE family PS00075 for example, the use of Greedy EM results in sensitivity 94.44% with specificity 100% for E-value threshold = 0.001. When using the MEME motifs, for the same classification behavior (specificity 100%) only 28 true positives are found, corresponding to sensitivity 38.89%, for E-value threshold = 10^{-6}. In all the ROC_{50} plots in Figure 4, the curves that correspond to Greedy EM–MAST method are located higher than those of MEME–MAST especially for lower E-values, thus indicating superior classification accuracy (higher sensitivity and specificity rates).

4 CONCLUSIONS

In this paper we have proposed a greedy EM algorithm for solving the multiple motif discovery problem in biological sequences. Our approach describes the problem through likelihood maximization by mixture learning using the EM algorithm. It learns a mixture of motifs model in a greedy fashion by iteratively adding new components. This is achieved through a combined scheme of local and global search which ensures fine tuning of the parameter vector of the new component. In addition a hierarchical clustering procedure is proposed based on the notion of kd-trees, which results in partitioning the (usually) large datasets (containing all substrings of length W) into a remarkable smaller number of candidate motif-models used for global searching. As it has been experimentally shown, this partitioning technique constitutes an effective strategy which manages to significantly reduce the time complexity for global searching without affecting the performance of the whole algorithm.

We have studied the performance of the proposed algorithm in several artificial and real biological datasets, including hard problems of almost indiscernible motif instances. Comparative results have also been provided through the application of the MEME approach which exhibits analogies to our method providing also an iterative algorithm of learning mixture models. The differences between the two approaches have been highlighted throughout this paper, while experiments have shown the superiority of the Greedy EM in discovering larger sets of more distinguishable (clearer) motifs (fingerprints) as suggested by the information content measure. The results obtained from the experimental study with the PROSITE database have also proved the ability of the greedy method to build more compact groups of diagnostic motifs for protein families that can provide with better homology searching and classification capabilities.

It must be noted that our approach has been developed mainly in an attempt to overcome some limitations of the MEME scheme, such as erasing input data each time a new motif is discovered using the assumption that this motif is correct, and limiting the model exclusively to the two-component case. Our technique actually overcomes these limitations based on recent methods for incremental mixture density estimation.

Ongoing research is mainly focused on working with multiple motifs of variable length. This can be viewed as a problem of expanding an existing model and determining the correct number of its parameters (the optimum width of the motif). Several model selection techniques can be adopted for this reason that have been proposed mainly for Gaussian mixture models, such as the likelihood ratio test (LRT), the minimum description length (MDL), the Markov chain Monte Carlo (MCMC) method, the Bayesian information criterion (BIC), the asymptotic information criterion (AIC) and some recent Bayesian approaches (Roberts *et al.*, 1998; McLachlan and Peel, 2001).

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous referees for the valuable comments and critics that significantly improve the quality and presentation of this work.

REFERENCES

Attwood,T.K., Croning,M.D.R., Flower,D.R., Lewis,A.P., Mabey,J.E., Scordis,P., Selley,J. and Wright,W. (2000) PRINT-S: the database formerly known as PRINTS. *Nucleic Acids Res.*, **28**, 225–227.

Bailey,T.L. (1995) *Discovering motifs in DNA and protein sequences: the appoximate common substring problem*, PhD thesis, University of California, San Diego.

Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, **21**, 51–83.

Bailey,T.L. and Gribskov,M. (1998) Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, **14**, 48–54.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, **28**, 45–48.

Bentley,J.L. (1975) Multidimensional binary search trees used for associative searching. *Commun. ACM*, **18**, 509–517.

Berejano,G. and Yona,G. (2001) Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. *Bioinformatics*, **17**, 23–43.

Brazma,A., Jonasses,L., Eidhammer,I. and Gilbert,D. (1998) Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.*, **5**, 277–303.

Brejova,B., DiMarco,C., Vinar,T., Hidalgo,S.R., Holguin,C. and Patten,C. (2000) Finding patterns in biological sequences. *Project Report for CS798g*. University of Waterloo.

Califano,A. (2000) SPLASH: structural pattern localization and analysis by sequential histograms. *Bioinformatics*, **16**, 341–357.

Dempster,A.P., Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, **39**, 1–38.

Durbin,R., Eddy,S., Krough,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acid*. Cambridge University Press.

Gribskov,M. and Robinson,N.L. (1996) The use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Hertz,C.Z. and Stormo,G.D. (1999) Indentifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.

Hofmann,K., Bucher,P., Falquet,L. and Bairoch,A. (1999) The PROSITE database, its status in 1999. *Nucleic Acids Res.*, **27**, 215–219.

Hughey,R. and Krogh,A. (1998) SAM: sequence alignment and modeling software system. *Technical Report UCSC-CRL-96-22*. University of California, Santa Cruz, CA.

Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwland,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **226**, 208–214.

Li,J.Q. and Barron,A.R. (2000) Mixture density estimation. *In Advances in Neural Information Processing Systems*. The MIT Press, **12**, pp. 279–285.

McLachlan,G.M. and Peel,D. (2001) *Finite Mixture Models*. Wiley, New York.

Render,R.A. and Walker,H.F. (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, **26**, 195–239.

Rigoutsos,I. and Floratos,A. (1998) Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, **14**, 55–67.

Rigoutsos,L., Floratos,A., Parida,L., Gao,Y. and Platt,D. (2000) The emergency of pattern discovery techniques in computational biology. *Metabolic Engineering*, **2**, 159–177.

Roberts,S.J., Husmcier,D., Rezek,I. and Penny,W. (1998) Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 1133–1142.

Smola,A.J., Mangasarian,O.L. and Schölkopf,B. (1999) Space kernel feature analysis. *Technical Report, Data Mining Institute*. University of Wisconsin, Madison.

Sproull,R.F. (1991) Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica*, **6**, 579–589.

Tompa,M. (1999) An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. In *7th International Conference on Intelligent Systems for Molecular Biology*. Heidelberg, Germany, pp. 262–271.

Verbeek,J.J., Vlassis,N. and Kröse,B. (2001) Efficient greedy learning of Gaussian mixture. In *Proceedings of the 13th Belgium-Dutch Conference on Artificial Intelligence BNAIC'01*. Amsterdam, The Netherlands, pp. 251–258.

Vlassis,N. and Likas,A. (2002) A greedy EM algorithm for Gausian mixture learning. *Neural Processing Letters*, **15**, 77–87.