From: Proceedings of the Second International Conference on
Intelligent Systems for Molecular Biology
August, 1994, pp. 28-36.
AAAI Press

Fitting a mixture model by expectation maximization to discover motifs in biopolymers

Timothy L. Bailey[*] and **Charles Elkan**[†]
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, California 92093-0114
tbailey@cs.ucsd.edu and elkan@cs.ucsd.edu

Abstract

The algorithm described in this paper discovers one or more motifs in a collection of DNA or protein sequences by using the technique of expectation maximization to fit a two-component finite mixture model to the set of sequences. Multiple motifs are found by fitting a mixture model to the data, probabilistically erasing the occurrences of the motif thus found, and repeating the process to find successive motifs. The algorithm requires only a set of unaligned sequences and a number specifying the width of the motifs as input. It returns a model of each motif and a threshold which together can be used as a Bayes-optimal classifier for searching for occurrences of the motif in other databases. The algorithm estimates how many times each motif occurs in each sequence in the dataset and outputs an alignment of the occurrences of the motif. The algorithm is capable of discovering several different motifs with differing numbers of occurrences in a single dataset.

Introduction

Finding a cluster of numerous, similar subsequences in a set of biopolymer sequences is evidence that the subsequences occur not by chance but because they share some biological function. For example, the shared biological function which accounts for the similarity of a subset of subsequences might be a common protein binding site or splice junction in the case of DNA sequences, or the active site of related enzymes in the case of protein sequences. This paper describes an algorithm called MM which, given a dataset of unaligned, possibly related biopolymer sequences, estimates the parameters of a probabilistic model which could have generated the dataset. The probabilistic model is a two-component finite mixture model. One component describes a set of similar subsequences of fixed width (the "motif"), while the other component describes all other positions in the sequences (the "background"). Fitting the model to the dataset includes estimating the relative frequency of motif occurrences. This estimated frequency determines the threshold for a Bayes-optimal classifier that can be used to find occurrences of the motif in other databases. The motifs found by MM resemble profiles without gaps (Gribskov, Luthy, & Eisenberg 1990).

The MM algorithm is an extension of the expectation maximization technique for fitting finite mixture models developed by Aitkin & Rubin (1985). It is related to the algorithm based on expectation maximization described by Lawrence & Reilly (1990), but it relaxes the assumption that each sequence in the dataset contains one occurrence of the motif. Sequences containing zero, one or many occurrences of a motif can be modelled equally well by the model used by MM. In other words, MM solves an unsupervised learning problem: it is intended to be useful for discovering new motifs in datasets, treating each subsequence of a fixed width in the dataset as an unlabeled sample.

The MM algorithm has been implemented as an option to the MEME software for discovering multiple motifs in biopolymer sequences (Bailey & Elkan 1993). MM can therefore be used to discover multiple motifs in a dataset. Briefly, this is done by repeatedly applying MM to the dataset and then probabilistically erasing all occurrences of the discovered motif. Because MM estimates the number of occurrences of each motif, MEME using MM is able to find motifs with different numbers of occurrences in a single dataset. This increases the usefulness of MEME as a tool for exploring datasets that contain more than one motif.

The rest of this paper is organized as follows. Section 2 explains the finite mixture model used by MM, and Section 3 summarizes the analysis needed to apply the expectation maximization idea to this type of model. Section 4 describes the implementation of MM in the context of MEME. Section 5 presents experimental results of using the MM algorithm to discover motifs in several DNA and protein datasets. Finally, Section 6 concludes the paper by discussing the strengths and limitations of the MM algorithm.

[*] Supported by NIH Genome Analysis Pre-Doctoral Training Grant No. HG00005.

[†] Supported in part by the National Science Foundation under Award No. IRI-9110813.

The finite mixture model

The MM algorithm searches for maximum likelihood estimates of the parameters of a finite mixture model which could have generated a given dataset of biopolymer sequences. We will refer to the dataset as $Y = \langle Y_1, Y_2, \ldots Y_N \rangle$, where N is the number of sequences in the dataset. The sequences Y_i are assumed to be over some fixed alphabet, say $A = \langle a_1, a_2, \ldots, a_L \rangle$, which is given as input to the algorithm. The mixture model used by MM does not actually model the dataset directly. Instead, the dataset is broken up conceptually into all n (overlapping) subsequences of length W which it contains. This new dataset will be referred to as $X = \langle X_1, X_2, \ldots, X_n \rangle$. MM learns a finite mixture model which models the new dataset. Although this model does not, strictly speaking, model the original dataset, in practice it is a good approximation, especially when care is taken to ensure that the model does not predict that two overlapping subsequences in the new dataset both were generated by the motif. This is done by enforcing a constraint on the estimated probabilities of overlapping subsequences being motif occurrences. (How this constraint is enforced is discussed in Section 4.)

The model for the new dataset consists of two components which model the motif and background (non-motif) subsequences respectively. The motif model used by MM says that each position in a subsequence which is an occurrence of the motif is generated by an independent random variable describing a multinomial trial with parameter $f_i = (f_{i1}, \ldots, f_{iL})$. That is, the probability of letter a_j appearing in position i in the motif is f_{ij}. The parameters f_{ij} for $i = 1, \ldots, W$ and $j = 1, \ldots, L$ must be estimated from the data. The background model says that each position in a subsequence which is not part of a motif occurrence is generated independently, by a multinomial trial random variable with a common parameter $f_0 = (f_{01}, \ldots, f_{0L})$. In other words, MM assumes that a sequence of length W generated by the background model is a sequence of W independent samples from a single background distribution. The overall model for the new dataset which MM uses is that the motif model (with probability λ_1) or the background model (with probability $\lambda_2 = 1 - \lambda_1$) is chosen by nature and then a sequence of length W is generated according to the probability distribution governing the model chosen. In summary, the parameters for the overall model of the data assumed by MM are the mixing parameter $\lambda = (\lambda_1, \lambda_2)$, vectors of letter frequencies for the motif model $\theta_1 = (f_1, f_2, \ldots, f_W)$, and a single vector of letter frequences for the background model $\theta_2 = f_0$.

Expectation maximization in finite mixture models

The MM algorithm does maximum likelihood estimation: its objective is to discover those values of the parameters of the overall model which maximize the likelihood of the data. To do this, the expectation maximization algorithm (EM) for finite mixture models of Aitkin & Rubin (1985) is used. This iterative procedure finds values for $\lambda = (\lambda_1, \lambda_2)$ and $\theta = (\theta_1, \theta_2)$ which (locally) maximize the likelihood of the data given the model.

A finite mixture model assumes that data $X = \langle X_1, X_2, \ldots, X_n \rangle$ arises from two or more groups with known distributional forms but different, unknown parameters. The EM algorithm makes use of the concept of missing data. In this case, the missing data is the the knowledge of which group each sample in the data came from. The following notation is useful:

$$\begin{aligned} Z &= \langle Z_1, Z_2, \ldots, Z_n \rangle, \\ &\quad \text{where } n \text{ is the number of samples} \\ Z_i &= \langle Z_{i1}, Z_{i2} \rangle, \\ Z_{ij} &= \begin{cases} 1 & \text{if } X_i \text{ from group } j \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The variable Z_i gives the group membership for the ith sample. In other words, if $Z_{ij} = 1$ then X_i has the distribution $p(X_i|\theta_j)$. The values of the Z_{ij} are unknown, and are treated by EM as missing information to be estimated along with the parameters θ and λ of the mixture model.

The likelihood of the model parameters θ and λ given the joint distribution of the data X and the missing data Z is defined as

$$L(\theta, \lambda | X, Z) = p(X, Z | \theta, \lambda). \tag{1}$$

It can be shown that the logarithm of the likelihood (*log likelihood*) is

$$\log L(\theta, \lambda | X, Z) = \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \log(p(X_i|\theta_j)\lambda_j). \tag{2}$$

The EM algorithm iteratively maximizes the expected *log likelihood* over the conditional distribution of the missing data, Z, given (a) the observed data, X, and (b) current estimates of parameters θ and λ. This is done by repeatedly applying the E-step and M-step of the algorithm as described below.

The E-step of EM finds the expected value of the *log likelihood* (2) over the values of the missing data Z, given the observed data, X, and the current parameter values $\theta = \theta^{(0)}$ and $\lambda = \lambda^{(0)}$. This can be shown to be

$$E[\log L(\theta, \lambda | X, Z)] = \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij}^{(0)} \log p(X_i|\theta_j) + \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij}^{(0)} \log \lambda_j \tag{3}$$

where

$$Z_{ij}^{(0)} = \frac{p(X_i|\theta_j^{(0)})\lambda_j^{(0)}}{\sum_{k=1}^{2} p(X_i|\theta_k^{(0)})\lambda_k^{(0)}}, \qquad (4)$$

for $i = 1, \ldots, n$, and $j = 1, 2$.

The M-step of EM maximizes (3) over θ and λ in order to find the next estimates for them, say $\theta^{(1)}$ and $\lambda^{(1)}$. The maximization over λ involves only the second term in (3)

$$\operatorname*{argmax}_{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij}^{(0)} \log \lambda_j$$

which has the solution

$$\lambda_j^{(1)} = \sum_{i=1}^{n} \frac{Z_{ij}^{(0)}}{n}, \qquad j = 1, 2. \qquad (5)$$

We can maximize over θ by maximizing the first term in (3) separately over each θ_j. To solve

$$\theta_j^{(1)} = \operatorname*{argmax}_{\theta_j} \sum_{i=1}^{n} Z_{ij}^{(0)} \log p(X_i|\theta_j), \qquad (6)$$

for $j = 1, 2$, we need to know the form of $p(X_i|\theta_j)$. The MM algorithm assumes that the distributions for class 1 (the motif) and class 2 (the background) are

$$p(X_i|\theta_1) = \prod_{j=1}^{W} \prod_{k=1}^{L} f_{jk}^{I(k,X_{ij})} \text{ and} \qquad (7)$$

$$p(X_i|\theta_2) = \prod_{j=1}^{W} \prod_{k=1}^{L} f_{0k}^{I(k,X_{ij})} \qquad (8)$$

where X_{ij} is the letter in the jth position of sample X_i and $I(k,a)$ is an indicator function which is 1 if and only if $a = a_k$. That is,

$$I(k,a) = \begin{cases} 1 & \text{if } a = a_k \\ 0 & \text{otherwise} \end{cases}$$

For $k = 1, \ldots, L$ let

$$c_{0k} = \sum_{i=1}^{n} \sum_{j=1}^{W} Z_{i2}^{(0)} I(k, X_{ij}) \text{ and} \qquad (9)$$

$$c_{jk} = \sum_{i=1}^{n} E_i Z_{i1}^{(0)} I(k, X_{ij}) \qquad (10)$$

for $j = 1, \ldots, W$. Then c_{0k} is the expected number of times letter a_k appears in positions generated by the background model in the data, and c_{jk} for $j = 1, \ldots, W$ is the expected number of times letter a_k appears at position j in occurrences of the motif in the data.[1] We reestimate θ by substituting (7) and (8) into the right-hand side of (6) yielding

$$\theta^{(1)} = (\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_W) = \operatorname*{argmax}_{\theta} \sum_{j=0}^{W} \sum_{k=1}^{L} c_{jk} \log f_{jk}. \qquad (11)$$

Therefore

$$\hat{f}_{jk} = \frac{c_{jk}}{\sum_{k=1}^{L} c_{jk}} \qquad (12)$$

for $j = 0, \ldots, W$ and $k = 1, \ldots, L$.

Estimating the parameters of a multinomial random variable by maximum likelihood is subject to boundary problems. If any letter frequency $\hat{f}_{ij}$ ever becomes 0, as is prone to happen in small datasets, its value can never change. Following Brown *et al.* (1993) and Lawrence *et al.* (1993), the equations above for $\hat{f}_{ij}$ are replaced by

$$\hat{f}_{ij} = \frac{c_{ij} + \beta_j}{\sum_{k=1}^{L} c_{ik} + \beta}, \qquad (13)$$

$i = 0, \ldots, W$, $j = 1, \ldots, L$, $\beta = \sum_{k=1}^{L} \beta_k$. This turns out to be equivalent to using the Bayes estimate for the value of θ under squared-error loss (SEL) (Santner & Duffy 1989) assuming that the prior distribution of each θ_j, $P(\theta_j)$, is a so-called Dirichlet distribution with parameter $\beta' = (\beta_1, \ldots, \beta_L)$. The value of β' must be chosen by the user depending on what information is available about the distribution of θ_j for motifs and for the background. The choice of β' will be discussed in the next section.

This completes the analysis of how expectation maximization can be applied to find the parameters of the mixture model assumed by MM.

Implementation of MM

The implementation of the MM algorithm is straightforward. Let l_i for $i = 1, \ldots, N$ be the lengths of the individual sequences in the dataset Y. The motif and background models are stored as an array of letter frequency vectors $\theta = f_0, \ldots, f_W$. The overlapping subsequences of length W in the dataset are numbered left-to-right and top-to-bottom from 1 to n. The $Z_{k1}^{(0)}$ for $k = 1, \ldots, n$ are stored in an array z_{ij} where $i = 1, \ldots, N$ and $j = 1, \ldots, l_i$ with z_{ij} holding the value of $Z_{k1}^{(0)}$ corresponding to the subsequence

[1] The factor E_i in the calculation of the motif counts is the "erasing factor" for that position in the data. (Erasing is mentioned in the introduction and further described in the next section.) The erasing factors vary between 1 and 0 and are set to 1 initially. After each pass, they are reduced by a factor between 0 and 1 representing the probability that the position is contained in an occurrence of the motif found on that pass. The counts for the background are not scaled by the erasing factors to make the values of the *log likelihood* function comparable among passes.

starting in column j of sequence Y_i in the dataset. MM repeatedly applies the E-step and the M-step of EM to update θ, λ and z until the change in θ (Euclidean distance) falls below a user-specified threshold (by default 10^{-6}) or a user-specified maximum number of iterations (by default 1000) is reached.

The E-step updates the z array using Equation (4) and the mapping just described between Z_{i1} and z_{ij}. The z_{ij} values for each sequence are then normalized to sum to at most 1 over any window of size W following Bailey & Elkan (1993):

$$\sum_{j=k}^{k+W-1} z_{ij} \leq 1,$$

for $i = 1, \ldots, N$ and $k = 1, \ldots, l_i - W$. This is done because otherwise there is a strong tendency for MM to converge to motif models that generate repeated strings of one or two letters like "AAAAAA" or "ATATAT" because the overlapping substrings in the new dataset are *not* independent.

The M-step reestimates λ and θ using Equations (5) and (13), respectively. The pseudo-counts $(\beta_1, \ldots, \beta_L)$ are set to $\beta\mu_i$, $i = 1, \ldots, L$, where β is a user specified parameter, and μ_i is the average frequency of letter a_i in the dataset.

The MM algorithm is implemented as an option to the MEME software for discovering multiple motifs in biopolymer sequences (Bailey & Elkan 1993). The version of MEME which uses MM will be be referred to as MEME+. MEME+ searches for the best starting point for MM by trying values of $\lambda^{(0)}$ between $\sqrt{N}/n$ and $1/(2W)$ and values of $\theta^{(0)}$ which are derived from subsequences in the dataset. To save execution time, the number of different values for $\theta^{(0)}$ tried by MEME+ depends on the value of $\lambda^{(0)}$ being tried. MEME+ uses a heuristic to tell if a potential starting point is "good," runs MM to convergence on the best starting point found, and "erases" the occurrences of the motif discovered. Erasing is accomplished by updating the erasing factors e_{ij}, (which are the E_k mentioned in the previous section, re-subscripted analogously to Z_{1k}), as follows:

$$e_{ij}^{(1)} = e_{ij}^{(0)} \prod_{k=j-W+1}^{j} (1 - z_{ik})$$

Since z_{ij} is can be viewed as an estimate of the probability that position j in sequence Y_i is the start of an occurrence of the motif just discovered, $e_{ij}^{(1)}$ is an estimate of the probability that the position is *not* contained in an occurrence of any motif found by the algorithm so far. (See (Bailey & Elkan 1994) for the details of MEME+ and the heuristics it uses.)

The output of MEME+ includes a log-odds matrix *spec* and a threshold value t for each motif found. Together these form a Bayes-optimal classifier (Duda & Hart 1973) for the "zero-one" loss function. The log-odds matrix has L rows and W columns and is calculated as $spec_{ij} = \log(\hat{f}_{ij}/\hat{f}_{0j})$ for $i = 1, \ldots, W$ and $j = 1, \ldots, L$. The threshold t is set to $t = \log((1-\lambda_1)/\lambda_1)$. To use *spec* and t as a classifier with a new dataset, each (overlapping) subsequence $x = \langle x_1, x_2, \ldots, x_n \rangle$ is given a score $s(x) = \sum_{j=1}^{W} \sum_{i=1}^{L} I(i, x_j) spec_{ij}$. It can be shown that $s(x) = \log(p(x|\theta_1)/p(x|\theta_2))$. Bayesian decision theory says to classify sample x as being an occurrence of the motif only if

$$\begin{aligned} s(x) &> \log(P(background)/P(motif)) \\ &= \log((1-\lambda_1)/\lambda_1) \\ &= t. \end{aligned}$$

The threshold for any other loss function can easily be found by scaling t. The scaled threshold should be $t + \log(r_{12} - r_{22})/(r_{21} - r_{11})$, where r_{ij} is the loss incurred for deciding class i when the correct class is j, and class 1 is the motif, class 2 the background.

The execution time of MEME+ is dominated by the search for good starting points. Testing a single starting point takes time $O(NMW)$, the execution time of one iteration of EM. Approximately $O(NM)$ starting points are tested. So finding a good starting point takes execution time $O((NM)^2W)$. Running MM to convergence tends to take time $O((NM)^2W)$ since the number of iterations of EM required tends to depend at worst linearly on the size of the dataset, NM (data not shown). The overall time complexity of one pass of MEME+ is thus $O((NM)^2W)$, *i.e.* quadratic in the size of the dataset and linear in the width of the motif.

Experimental Results

We studied the performance of MEME+ on a number of datasets with different characteristics. The datasets are summarized in Table 1. Three of the datasets consist of protein sequences and three consist of DNA sequences. Three contain a single known motif. One contains two known motifs, each of which occurs once in each sequence. One contains three known motifs, each of which occurs multiple times per sequence. And one contains two motifs, each of which occurs in only about half of the sequences.

The protein datasets, lipocalin, hth, and farn, are described in (Lawrence *et al.* 1993) and were used to test the Gibbs sampling algorithm described there. We reiterate briefly here. The lipocalin proteins bind small, hydrophobic ligands for a wide range of biological purposes. The dataset contains the five most divergent lipocalins with known 3D structure. The positions of the two motifs in each of the sequences in the lipocalin dataset are known from structural comparisons. The hth proteins contain occurrences of DNA-binding structures involved in gene regulation. The correct locations of occurrences of the motif are known from x-ray and nuclear magnetic resonance structures, or from substitution mutation experiments,

dataset name	*type*	*number of sequences*	*sequence length (avg)*	*W*	*motif name*	*sites*	
						proven	*total*
lipocalin	protein	5	182	16	lipA	5	5
					lipB	5	5
hth	protein	30	239	18	hth	30	30
farn	protein	5	380	12	farnA	none	30
					farnB	none	26
					farnL	none	28
crp	DNA	18	105	20	crp	18	24
lexa	DNA	16	200	20	lexa	11	21
crplexa	DNA	34	150	20	crp	18	25
					lexa	11	21

Table 1: Overview of the contents of the datasets. Proven sites are those which have been shown to be occurrences of the motif by laboratory experiment (*i.e.*, footprinting, mutagenesis or structural analysis). Total sites include the proven sites as well as sites that have been reported in the literature but whose identification was based primarily on sequence similarity with known sites (*i.e.*, "putative" sites).

or both. The farn dataset contains isoprenyl-protein transferases, essential components of the cytoplasmic signal transduction networks. No direct structural information is known for the proteins in the dataset, so we used the starting positions for the three motifs reported by Lawrence *et al.* (1993). These starting positions agreed with the results of earlier sequence analysis work (Boguski *et al.* 1992), (Boguski, Murray, & Powers 1992).

The three DNA datasets, crp, lexa and crplexa, are described in (Bailey & Elkan 1993) and were used to test MEME there. They contain DNA sequences from *E. coli.* The crp dataset contains known binding sites for CRP (Lawrence & Reilly 1990), and a few sites which have been identified by similarity to the known motif. The lexa dataset sequences contain known binding sites for LexA (Hertz, Hartzell, III, & Stormo 1990), and some that have been identified by similarity to known sites. The crplexa dataset contains all the sequences in the crp and lexa datasets.

To evaluate the success of MEME+, we ran it on each dataset, derived Bayes-optimal classifiers from the motif models found, and used these classifiers to classify that dataset. In each dataset, the predicted occurrences of each discovered motif were compared with the proven and putative occurrences of the known motifs. Success was measured using *recall* defined as tp/p and *precision* defined as $tp/(tp + fp)$. Here, p is the number of occurrences of a known motif in the dataset ("positives"), tp is the number of correctly classified positives ("true positives"), and fp is the number of non-occurrences classified as occurrences ("false positives"). These statistics can be used as estimators of the true precision and recall of the motif learned by MEME+ if it is used to find occurrences of the motif in a different dataset.[2]

Table 2 shows the results of running MEME+ on the datasets and analyzing the motifs produced. MEME+ finds all the known motifs. The *recall* and *precision* of all the discovered motifs is quite high except for the lipocalin dataset. With one exception (farnB), MEME+ finds motifs with similar or higher likelihood than the known motifs indicating that MEME+ is not getting stuck at local optima. This also indicates that, based on statistical evidence alone, the known motifs may not always be the most significant patterns in the datasets. The low *precision* of the lipocalin motifs turns out to be because MEME+ finds a motif on the first pass which combines the two known motifs and has much higher *log likelihood* than either of the known motifs alone. Since MEME+ is doing maximum likelihood estimation, if a combination of two motifs is more statistically significant than either of the motifs alone, this behavior is to be expected. Fortunately, versions of subtle motifs like those in the lipocalin dataset will still be found, though they may tend to be over-general and have poor *precision*.

It is also notable that the known motifs tend to be found on the first passes of MEME+, indicating that it will be a reliable tool for discovering unknown motifs. Additional passes of MEME+ were run on the datasets, and the *log likelihood* statistic tended to be much lower than for the motifs shown in Table 2 (data not shown). In searching for unknown motifs, this would be evidence that motifs discovered in earlier passes are significant from a biological (or at least statistical) point of view.

[2] Each discovered motif was compared with each motif known to occur in the dataset. *recall* and *precision* are relative to the "closest" known motif where "closest" means highest *recall*. The comparison between each discovered motif and each known motif was done once for each possible shifting of the known motif a fixed number of positions, i, $|i| \leq \lfloor W/2 \rfloor$. MEME+ was thus credited with finding a motif even if the predicted occurrences were displaced a small, fixed amount.

	Output of MEME+		*Analysis of discovered motifs*				
dataset name	*pass*	*log likelihood of discovered motif* (d)	*motif name*	*recall*	*precision*	*log likelihood of known motif* (k)	*difference* $(d-k)$
lipocalin	1	-55013	lipA	1.000	0.357	-55090	77
	2	-55057	lipB	0.400	0.200	-55092	35
hth	1	-496332	hth	0.933	0.571	-496346	14
farn	1	-92518	farnL	0.917	0.880	-92525	7
	2	-92585	farnB	0.615	0.842	-92517	-68
	3	-92569	farnA	0.733	0.647	-92566	-3
crp	1	-60547	crp	0.792	0.905	-60590	43
lexa	1	-109155	lexa	0.842	0.615	-109147	-8
crplexa	1	-169923	lexa	0.842	0.696	-169918	-5
	2	-170048	crp	0.667	0.471	-170116	68

Table 2: Overview of results of MEME+ on test datasets. MEME+ was run with W set to the values shown in Table 1 and $\beta = 0.01$. The *log likelihood* values are base-2 logarithms.

Discussion

The MM algorithm and its implementation in MEME+ have several important advantages over previously reported algorithms that perform similar tasks. This section explains these advantages, and then discusses several limitations of MM and MEME+ the lifting of which would increase their usefulness for exploring collections of DNA and protein sequences.

The Gibbs sampling algorithm of Lawrence *et al.* (1993) is the most successful existing general algorithm for discovering motifs in sets of biosequences. MM has two major advantages over this algorithm. First, MM does not require input sequences to be classified in advance by a biologist as known to contain the motif that is being searched for. Instead, MM estimates from the data how many times a motif appears. This capability is quite robust: experiments show that even when only 20% of the sequences in a dataset contain a motif, the motif can still be characterized well (data not shown). Second, MM uses a formal probabilistic model of the entire input dataset, and systematically selects values for the parameters of this model that maximize the likelihood of the model. The MM model allows us to compare in a principled way the motif characterizations discovered by MEME+ and characterizations obtained by other methods. In most cases, the characterizations discovered by MEME have higher likelihood.

As pointed out by Lawrence *et al.* (1993) and by others, the fundamental practical difficulty in discovering motifs is the existence of many local optima in the search space of alternative motif models. The MM algorithm, like all expectation maximization applications, is a gradient descent method that cannot escape from a local optimum. The MEME+ implementation of MM uses several heuristics to overcome the problem of local optima. These heuristics are all variations on a common theme, and should be useful in other applications also. The theme is to search the space of possible starting points for gradient descent systematically. By contrast, Gibbs sampling algorithms combine gradient search steps with random jumps in the search space. These algorithms can spend an unpredictable number of iterations on a "plateau" before converging, whereas MM always converges in a predictable, relatively small number of iterations.

The focus of our current research is to overcome two significant limitations of MM and MEME+. The first of these is that all motifs found are constrained to have the same width, which is a parameter specified by the user. The main obstacle to estimating motif width endogenously is that likelihood values are not comparable for models that assume different motif widths.

The second limitation is that the number of different motifs present in a dataset is not estimated by the algorithm. We plan to overcome this limitation by generalizing from a two component mixture model to models with multiple components. A deep difficulty with multiple component models is that the induced search space is of even higher dimensionality than with two components, and local optima are even more pesky. Our current intention is to use the results of MEME+ as starting points for fitting models with multiple components. Doing this should have the additional benefit of allowing similar motifs discovered in different passes of MEME+ to be merged if overall likelihood is thereby increased.

Another possible area for enhancement of the algorithm is to allow the user to specify a "weight" for each sequence to reduce any tendency to find motifs skewed towards species which may be overrepresented in the input dataset. For example, if a dataset contained several sequences known to be closely related evolutionarily, they could be given weights less than 1, while more distantly related sequences could be given unit weights. These weights could be used in the EM algorithm when estimating the letter frequencies of the motif in such a way to prevent the closely related sequences from dominating.

Acknowledgements

Michael Gribskov and Douglas W. Smith provided valuable insights and comments during the course of this research.

References

Aitkin, M., and Rubin, D. B. 1985. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society, B* 47(1):67–75.

Bailey, T. L., and Elkan, C. 1993. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Technical Report CS93-302, Department of Computer Science, University of California, San Diego.

Bailey, T. L., and Elkan, C. 1994. Fitting a two-component mixture model by expectation maximization to discover motifs in biopolymers. Technical Report CS94-351, Department of Computer Science, University of California, San Diego.

Boguski, M. S.; Hardison, R. C.; Schwartz, S.; and Miller, W. 1992. Analysis of conserved domains and sequence motifs in cellular regulatory proteins and locus control regions using new software tools for multiple alignment and visualization. *New Biologist* 4(3):247–260.

Boguski, M. S.; Murray, A. W.; and Powers, S. 1992. Novel repetitive sequence motifs in the alpha and beta subunits of prenyl-protein transferases and homology of the alpha subunit to the MAD2 gene product of yeast. *New Biologist* 4(4):408–411.

Brown, M.; Hughey, R.; Krogh, A.; Mian, I. S.; Sjolander, K.; and Haussler, D. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In *Intelligent Systems for Molecular Biology*, 47–55. AAAI Press.

Duda, R. O., and Hart, P. E. 1973. *Pattern Classification and Scene Analysis.* John Wiley & Sons, Inc.

Gribskov, M.; Luthy, R.; and Eisenberg, D. 1990. Profile analysis. *Methods in Enzymology* 183:146–159.

Hertz, G. Z.; Hartzell, III, G. W.; and Stormo, G. D. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computer Applications in Biosciences* 6(2):81–92.

Lawrence, C. E., and Reilly, A. A. 1990. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *PROTEINS: Structure Function and Genetics* 7:41–51.

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

Santner, T. J., and Duffy, D. E. 1989. *The Statistical Analysis of Discrete Data.* Springer Verlag.