Machine Learning, 21, 51–80 (1995)

Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization

TIMOTHY L. BAILEY tbailey@cs.ucsd.edu

CHARLES ELKAN elkan@cs.ucsd.edu

Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California 92093-0114

Editors: Lawrence Hunter, David Searls, and Jude Shavlik

Abstract. The MEME algorithm extends the expectation maximization (EM) algorithm for identifying motifs in unaligned biopolymer sequences. The aim of MEME is to discover new motifs in a set of biopolymer sequences where little or nothing is known in advance about any motifs that may be present. MEME innovations expand the range of problems which can be solved using EM and increase the chance of finding good solutions. First, subsequences which actually occur in the biopolymer sequences are used as starting points for the EM algorithm to increase the probability of finding globally optimal motifs. Second, the assumption that each sequence contains exactly one occurrence of the shared motif is removed. This allows multiple appearances of a motif to occur in any sequence and permits the algorithm to ignore sequences with no appearance of the shared motif, increasing its resistance to noisy data. Third, a method for probabilistically erasing shared motifs after they are found is incorporated so that several distinct motifs can be found in the same set of sequences, both when different motifs appear in different sequences and when a single sequence may contain multiple motifs. Experiments show that MEME can discover both the CRP and LexA binding sites from a set of sequences which contain one or both sites, and that MEME can discover both the -10 and -35 promoter regions in a set of *E. coli* sequences.

Keywords: Unsupervised learning, expectation maximization, consensus sequence, motif, biopolymer, promoter, binding site, DNA, protein, sequence analysis

1. Introduction

The problem addressed by this work is that of identifying and characterizing shared motifs in a set of unaligned genetic or protein sequences. A motif is defined here as a pattern common to a set of nucleic or amino acid subsequences which share some biological property of interest such as being DNA binding sites for a regulatory protein. In computer science terminology, the problem is, given a set of strings, to find a set of non-overlapping, approximately matching substrings. In this report we are concerned only with *contiguous* motifs. In biological terms, this means that appearances of a motif may differ in point mutations, but insertions or deletions are not allowed. In computer science terms, this means that the approximately matching substrings must all have the same length. A simpler version of the problem is, given a dataset of biopolymer sequences believed to contain a single shared motif, to locate the starting position in each sequence of the appearance of the shared motif and to describe the shared motif. This report addresses the more general problem of finding and describing multiple, distinct shared motifs in a set of biopolymer sequences. It is not assumed that anything is known

in advance about the width, position or letter frequencies of the motifs, or even how many common motifs may exist in a set of sequences.

Several methods have been presented in the literature which work on problems related to discovering multiple, distinct shared motifs in a set of biological sequences. The purpose of this research is to extend the range of problems that can be attacked. Hertz *et al.* (1990) presented a greedy algorithm for discovering a single, shared motif that is present once in each of a set of sequences. Lawrence and Reilly (1990) extended that work by developing an expectation maximization (EM) algorithm for solving the same problem. Lawrence *et al.* (1993) solve the related problem of discovering multiple, distinct motifs when the number of occurrences of each motif in each sequence is known using a Gibbs sampling strategy.

This report describes MEME, a new tool intended to help discover motifs when neither the number of motifs nor the number of occurrences of each motif in each sequence is known.[1] MEME incorporates three novel ideas for discovering motifs.

- First, subsequences which actually occur in the input DNA or protein sequences are used as the starting points from which EM converges iteratively to locally optimal motifs. This increases the likelihood of finding globally optimal motifs.
- Second, a heuristic modification of the EM algorithm allows the assumption that each sequence contains exactly one occurrence of the shared motif to be removed. This allows multiple appearances of a motif to occur in any sequence and permits the algorithm to ignore sequences with no appearance of a shared motif, which increases its resistance to noisy data.
- Third, motifs are probabilistically erased after they are found. This allows several distinct motifs to be found in the same set of sequences, both when different motifs appear in different sequences and when a single sequence may contain multiple motifs.

1.1. Searching tools versus learning tools

This section explains the place of MEME, in the spectrum of sequence analysis tools. Experts on biological sequence analysis may wish to skip directly to the next section.

Searching tools. Sequence analysis tools may be divided into two broad categories, searching tools and learning tools. GRAIL, BLASTX, FASTA, etc. are searching tools, whereas MEME is a learning tool. A searching tool (also called a pattern-matching tool) takes as input one or more sequences and a pattern, and decides if the pattern matches each input sequence, and if so, where. The pattern may be (i) another sequence, as with BLASTX and FASTA, (ii) a consensus subsequence or regular expression defining a motif, as with ProSearch (Kolakowski, et al., 1992), or (iii) a more high-level combination of features, as with GRAIL (Uberbacher & Mural, 1991).

Learning tools. A supervised learning tool (also called a supervised pattern-recognition tool) takes as input a set of sequences, and *discovers* a pattern that all the sequences share. Supervised learning is often done by humans rather than by software, because

it is an open-ended problem that is harder than searching. For example, the PROSITE profiles were created by Amos Bairoch by personally examining families of proteins (Bairoch, 1993).

An unsupervised learning tool takes as input a set of sequences, and discovers a pattern that *some* of the sequences share. Unsupervised learning is harder than supervised learning because the space of possible patterns is much larger. The pattern to be discovered is not required to be in any given input sequence, so the unsupervised learning algorithm must simultaneously look for a cluster of input sequences and a pattern that the members of this cluster do have in common. MEME performs unsupervised learning.

The output of a learning tool, namely a pattern, is often given to a search tool in order to find new sequences that exhibit the pattern. However, even if all the members of a family of sequences are already known, applying a learning tool to the family can still be useful, because examining the patterns that subsets of the family have in common can give insight into structure, function, and evolution.

1.2. The expectation maximization (EM) algorithm

Lawrence and Reilly (Lawrence & Reilly, 1990) introduced the expectation maximization method as a means of solving a supervised motif learning problem. Their algorithm takes as input a set of unaligned sequences and a motif length (W) and returns a probabilistic model of the shared motif. The idea behind the method is that each sequence in the dataset contains a single example of the motif. We shall refer to this model of the data as the 'one-occurrence-per-sequence' model or just the 'one-per' model. It is assumed that where the motif appears (what its starting offset is) in each example is unknown. If this were known, subsequences of length W from each sequence starting at the known offset could be aligned, since no insertions or deletions are allowed, and the observed frequencies of the letters in each column of the alignment could be used as a model of the motif.

In fact, if each example of the motif is assumed to have been generated by a sequence of independent, discrete random variables, then the observed frequencies of the letters in the columns are the maximum likelihood estimates of the distributions of the random variables. Of course, since the original sequences in the dataset are unaligned, the offsets are not known, so they must also be estimated. To do this, the EM algorithm estimates the probability that the shared motif starts in position j in sequence i in the dataset, given the data and an initial guess at a description of the motif. These probability estimates, $\hat{z}_{ij}$, are then used to reestimate the probability of letter l in column c of the motif, ρ_{lc}, for each letter in the alphabet and $1 \leq c \leq W$. How the reestimations are done is described in the Appendix. The EM algorithm alternately reestimates z and ρ until ρ changes very little from iteration to iteration. (The notation z is used to refer to the matrix of offset probabilities z_{ij}. Likewise, ρ refers to the matrix of letter probabilities ρ_{ij}.)

A pseudo-code description of the basic EM algorithm is given below. EM starts from an estimate of the model parameters, ρ, provided by the user or generated at random.

```
1. EM (dataset, W) {
2.     choose starting point (ρ)
3.     do {
4.         reestimate z from ρ
5.         reestimate ρ from z
6.     } until (change in ρ < ε)
7.     return
8. }
```

The EM algorithm simultaneously discovers a model of the motif (the sequence of independent discrete random variables with parameters ρ) and estimates the probability of each possible starting point of examples of the motif in the sequences in the dataset (z). By definition (Duda & Hart, 1973), the likelihood of the model given the training data is the probability of the data given the model. The EM algorithm finds values of the model parameters which maximize the expected likelihood of the data given the model ρ, and the missing data z. For the one-occurrence-per-sequence model of the data used by Lawrence and Reilly (1990), the logarithm of the likelihood is

$$\log(\textit{likelihood}) = N \sum_{j=1}^{W} \sum_{l \in \mathcal{L}} f_{lj} \log(\rho_{lj}) + N(L - W) \sum_{l \in \mathcal{L}} f_{l0} \log(\rho_{l0}) + N \log(\frac{1}{L - W + 1})$$

where N is the number of sequences in the dataset, L is the length of the sequences, W is the length of the shared motif, $\mathcal{L}$ is the alphabet of the sequences, ρ_{lj} is the (unknown) probability of letter l in position j of the motif, ρ_{l0} is the (unknown) probability of letter l in all non-motif positions, f_{lj} is the observed frequency of the letter l in position j of the motif, and f_{l0} is the observed l in all non-motif positions of the sequences.

It has been shown that expectation maximization algorithms find values for the model parameters at which the likelihood function assumes a local maximum (Dempster, et al., 1977). It is reasonable to assume that the correct solution to the problem of characterizing the shared motif occurs at the global maximum of the likelihood function. For this reason, all else being equal, parameter values for the model which give higher values of the likelihood function are considered better solutions to the problem.[2]

1.3. Limitations of EM and the one-occurrence-per-sequence model.

EM and the one-per model suffer from several limitations. First, it is not clear how to choose a starting point (an initial value of ρ) nor when to quit trying different starting points. This makes it difficult to be satisfied that the correct shared motif has been found. Second, the one-per model assumes that each sequence in the dataset contains exactly one appearance of the shared motif. This means that sequences with multiple appearances will under-contribute, and sequences with no appearances will over-contribute to the

characterization of the motif. Having many sequences with no appearances of the motif in the dataset may make it impossible for EM with the one-per model to find the shared motif at all. Finally, EM with the one-per model assumes that there is only one shared motif in the sequences, and does not keep looking for further motifs after characterizing one. This makes EM with the one-per model incapable of finding motifs with insertions of variable length and incapable of discovering multiple motifs that may occur in the same or different sequences in a given dataset. Eliminating or reducing these limitations of EM with the one-per model would make the method less susceptible to noise in the dataset, able to find more complex patterns in the data, and last but not least, useful for exploring datasets which may contain instances of several different motifs.

The algorithm described in this report, MEME, extends the EM algorithm to overcome the limitations described above. MEME chooses starting points systematically, based on all subsequences of sequences in the training dataset. It allows the use of either the one-per model or a different model which eliminates the assumption of one sequence/one occurrence and allows each sequence to contain zero, one or several appearances of the shared motif. We call this new model the 'n-occurrences-per-dataset' model or just the 'n-per' model. because the it assumes that the dataset contains exactly n occurrences of the motif, where n is specified by the user. Finally, MEME probabilistically erases the appearances of a motif after it is found, and continues searching for further shared motifs in the dataset.

The MEME algorithm with the n-per model was tested on two datasets. The first was a dataset combining 18 *E. coli* sequences containing CRP binding sites (Lawrence & Reilly, 1990) and 16 sequences containing LexA binding sites (Hertz, et al., 1990). MEME discovered the LexA binding site on its first pass and the CRP binding site on its second pass. The second dataset contained 231 *E. coli* promoter sequences (Harley & Reynolds, 1987).[3] MEME discovered the TATAAT and TTGACA consensus sequences[4] on the first and second passes, respectively. This demonstrates the ability of MEME to avoid local optima, to tolerate large number of sequences which do not contain the motif, and to find multiple motifs in a single dataset.

2. The MEME algorithm

The MEME algorithm has at its core a modified version of the EM algorithm (Lawrence & Reilly, 1990). The pseudo-code for the algorithm is given below. In the inner loop, an algorithm based on the EM algorithm is run repeatedly with different starting points for the chosen model (either one-per model or n-per model). We shall refer to this particular application of the EM algorithm as simply 'EM' in what follows. The starting points are derived from actual subsequences which occur in the input dataset. EM is run only one iteration, not to convergence, from each starting point to save time. Each run of EM produces a probabilistic model of a possible shared motif. The starting point which yields the model with the highest likelihood is chosen and EM is run to convergence from this starting point. The model of the shared motif thus discovered is printed. Finally, all appearances of the shared motif in the dataset are erased. The outer loop repeats the

whole process to discover further shared motifs. The following sections describe each of these steps of the algorithm in more detail.

```
1.  MEME (dataset, W, NSITES, PASSES) {
2.      for i = 1 to PASSES {
3.          for each subsequence in dataset {
4.              run EM for 1 iteration with starting point
5.                  derived from this subsequence
6.              choose model of shared motif with highest likelihood
7.              run EM to convergence from starting point
8.                  which generated that model
9.              print converged model of shared motif
10.             erase appearances of shared motif from dataset
11.         }
12.     }
13. }
```

The output of MEME includes a specificity or log-odds matrix, *spec*. The log-odds matrix has L rows and W columns and is calculated as $spec_{ij} = \log(\hat{\rho}_{ij}/\hat{\rho}_{0j})$ for $i \in \mathcal{L}$ and $j = 1, \ldots, W$. The information content score of a subsequence is calculated by summing the entries in the matrix corresponding to the letters in the subsequence.[5] This score gives a measure of the likelihood of the subsequence being an instance of the motif versus an instance of the "background". Together with a suitable threshold, the information content score can be used to classify subsequences in new sequences not part of the training set.

2.1. *Using subsequences as starting points for EM*

Given different starting points (i.e., initial letter probability matrices ρ) the EM algorithm may converge to different final models. These models are local maxima of the likelihood function described earlier. The correct model for the shared motif is expected to be the model which globally maximizes the likelihood function, but EM is not guaranteed to find the global maximum, only a local maximum. Previous authors (Lawrence & Reilly, 1990; Cardon & Stormo, 1992) have recommended using several starting points for EM and choosing the model with the highest likelihood, but how to choose the starting points has not been discussed in detail.

One might try using randomly chosen letter frequency matrices as starting points, but the sequences in the dataset provide a way to choose more intelligent ones. Since our models for motifs do not allow for insertions or deletions, the optimal model must agree very well with some contiguous subsequences of the sequences in the dataset—the instances of the motif in the sequences. A good way to search the space of possible starting points for EM should thus be to convert each subsequence of length W into a letter

probability matrix and use each such matrix as a starting point. This is the approach used by MEME. Since the starting point letter frequency matrices obtained from subsequences corresponding to the actual occurrences of the shared motif should be "close" to the correct letter probability matrix (i.e., model), EM should tend to converge to the global optimum when run with them as starting points.[6]

For example, suppose the unknown optimal value of ρ for the shared motif that we are trying to discover using MEME is actually

letter	*position in motif*					
	1	2	3	4	5	6
A	0.1	0.8	0.1	0.5	0.6	0.1
C	0.1	0.1	0.1	0.3	0.2	0.1
G	0.2	0.0	0.1	0.1	0.1	0.1
T	0.6	0.1	0.7	0.1	0.1	0.7

and the consensus sequence is TATAAT. Presumably, this sequence or something close to it (i.e., with few mutations) occurs in at least one of the sequences in the dataset. It is reasonable to postulate that if we choose as a starting point for EM a letter probability matrix derived in some simple manner from the consensus sequence, or a subsequence similar to it, then EM should tend to converge to the optimal model. If we try all of the subsequences (of length six in this example) of the sequences in the dataset, it is reasonable to assume that at least one of them will be "close" to TATAAT and will cause EM to converge to the optimal model. (Note that MEME does not use all possible subsequences of a given length, just the ones which actually *occur* in the dataset.)

The question remains of how to convert a subsequence into a letter probability matrix. One cannot simply convert it to a matrix with probability 1.0 for the letter in the subsequence and 0.0 for all others, i.e., convert TATAAT to

letter	*position in motif*					
	1	2	3	4	5	6
A	0.0	1.0	0.0	1.0	1.0	0.0
C	0.0	0.0	0.0	0.0	0.0	0.0
G	0.0	0.0	0.0	0.0	0.0	0.0
T	1.0	0.0	1.0	0.0	0.0	1.0

because the EM algorithm cannot move from such a starting point. With such a starting point, all offset probabilities will be estimated to be 0.0 except for subsequences which match the starting point subsequence exactly. This will cause reestimation of the letter frequencies to yield the starting point again.

An effective, if somewhat arbitrary solution is to fix the frequency of the letter in the subsequence at some value $0 < X < 1$, and fix the frequencies of the other letters at $(1-X)/(M-1)$ where M is the length of the alphabet. This ensures that the frequencies in each column sum to 1.0 and that, for X close to 1.0, the starting point is "close" to the subsequence. The results reported in this paper are for $X = 0.5$. Values of X between 0.4 and 0.8 worked approximately equally well (experimental data not shown). With this value of X, the starting point for EM generated from the subsequence TATAAT is

letter	*position in motif*					
	1	2	3	4	5	6
A	0.17	0.5	0.17	0.5	0.5	0.17
C	0.17	0.17	0.17	0.17	0.17	0.17
G	0.17	0.17	0.17	0.17	0.17	0.17
T	0.5	0.17	0.5	0.17	0.17	0.5

It would be highly expensive computationally to run EM until convergence from every possible starting point corresponding to some subsequence of length W in the input dataset. It turns out that this is not necessary. EM converges so quickly from subsequences which are similar to the shared motif that the best starting point can often be detected by running only one iteration of EM. As will be described below, MEME was able to find shared motifs when run for only one iteration from each possible subsequence starting point, and then run until convergence from the starting point with the highest likelihood. In other words, MEME runs EM for specified number of iterations (one iteration in all the results reported here) on each subsequence starting point, chooses the starting point that yields the highest likelihood, and then runs EM to convergence from this starting point.

Since each iteration of the EM algorithm takes computation time roughly linear in the size of the dataset, and the number of subsequences is linear in the size of the dataset, MEME takes time $O(n^2)$ where n is the size of the dataset in characters.

2.2. *Dealing with multiple appearances of a shared motif*

MEME allows the user to specify that either the one-per model or the n-per model be used. With the one-per model, MEME uses the EM algorithm of Lawrence and Reilly (Lawrence & Reilly, 1990) to fit the model to the dataset. To fit the n-per model, a heuristic modification of the EM algorithm is used.

The one-per model assumes that each sequence in the dataset contains exactly one appearance of the shared motif to be characterized. This assumption determines the way in which the offset probabilities are reestimated. The reestimation procedure ensures that the offset probabilities for each sequence sum to 1.0. This means that if a given sequence has more than one appearance of the shared motif, it cannot contribute any more to the reestimation of the letter frequencies than a sequence with only one appearance. Additionally, if a sequence has no appearances of the shared motif—a common event when exploring for new shared motifs—it contributes erroneously to the reestimation of the letter frequencies.

MEME modifies the EM algorithm a when fitting the n-per model to a dataset. Instead of normalizing the reestimated offset probabilities to sum to 1.0 for each sequence, all offset probabilities are normalized to sum to a user-supplied value *NSITES*, subject to the constraint that no single offset probability may exceed 1.0. This normalization is done over all sequences simultaneously, not sequence by sequence. The intent is for *NSITES* to be the expected number of appearances of the shared motif in the dataset. If *NSITES* is set equal to the number of sequences in the dataset, it is possible for the

n-per model to get approximately the same results as the one-per model on a dataset that has one appearance of the shared motif in each sequence. For datasets with the appearances of the motif distributed other than one per sequence, the MEME with the n-per model is able to choose models that assign the offset probabilities in any fashion which satisfies the two constraints mentioned above.

The relaxation of the one motif appearance per sequence constraint in the n-per model allows MEME to benefit from sequences with multiple appearances of the shared motif. It also can help alleviate the problem of sequences which do not contain the motif blurring its characterization. When *NSITES* is lower than the number of sequences in the dataset, MEME can assign very low offset probabilities to *all* positions in a sequence that does not contain the motif at all. By contrast, the one-per model must assign offset probabilities summing to 1.0 to each sequence in the dataset. The effect of various settings for *NSITES* is discussed in Section 4.3. In summary, the exact value chosen for *NSITES* is not critical, so it is not necessary to know in advance exactly how many times a motif is present in the dataset.

One side effect of allowing a single sequence to have offset probabilities that sum to more than 1.0 is that long repeated sequences are seen by MEME using the n-per model as though they were multiple appearances of a shorter sequence. For example, if W is 6, the sequence AAAAAAAA is treated by the n-per model roughly as though it were three appearances of the sequence AAAAAA. This is so because the n-per model might allow offsets 1, 2 and 3 of the sequence to have the maximum probability of 1.0. (The one-per model would not allow this, since the total offset probability for a single sequence must sum to 1.0.) This is problematic because it is far more surprising to find 3 *non-overlapping* occurrences of the sequence AAAAAA than to find one occurrence of sequence AAAAAAAA. So, we would like MEME to search for *NSITES* non-overlapping occurrences of the motif. To overcome this difficulty, MEME enforces an additional constraint when calculating the offset probabilities for the n-per model. It renormalizes the offset probabilities so that no W adjacent offsets have probabilities that sum to greater than 1.0. This essentially makes the n-per model treat sequences like AAAAAAAA the same way as the one-per model does, assigning at most probability 1/3 to each of the three offsets at which identical subsequences AAAAAA start.

2.3. *Finding several shared motifs*

When a single dataset of sequences contains more than one distinct shared motif, EM with the one-per model cannot directly find more than one of them. If the motifs have some similarity, EM may always converge to the most conserved motif.[7] Another possibility is that EM may converge to a model that describes part of the most conserved motif—its left or right side for instance. The MEME algorithm solves this problem by probabilistically erasing the shared motif found by EM and then repeating EM to find the next shared motif. By effectively removing each motif as it is found, MEME is able to find the next motif without interference from the more conserved motifs found first.

The manner in which MEME erases a motif is designed to be as continuous as possible. New variables w_{ij} are defined which associate a weight with position j in sequence i. The

weights represent the probability that the given position in the given sequence is *not* part of a motif previously discovered by MEME. The weights are all set initially to 1.0. After MEME discovers a shared motif, the offset probability z_{ij} gives the probability that an appearance of the motif starts at a position j in sequence i. So, assuming independence, the probability that position k in sequence i is *not* part of the newly discovered motif is the product of $(1 - z_{ij})$ for all j between $k - W$ and k. So the old value of w_{ij} is updated by multiplying it by the probability that no potential motif which overlaps it is an example of the newly discovered shared motif.

The w_{ij} are used in reestimating the letter frequencies. Instead of summing the offset probabilities z_{ij}, the weighted offset probabilities $w_{ij} \cdot z_{ij}$ are summed. To understand how the weighting scheme effectively erases previously discovered motifs, suppose that MEME has discovered one motif and is looking for the second. Suppose position j in sequence i was the start of an appearance of the first motif found. Then the new weights w_{ij} through $w_{i,(j+W-1)}$ will all be less than $1 - z_{ij}$. Hence they cannot contribute much to the reestimation of ρ and are effectively erased. Notice that if a position only matches the discovered motif poorly, then z_{ij} will be low, so the weight for that position will remain fairly high. The degree to which a position is erased is proportional to the certainty (z_{ij}) that it is part of a previously discovered motif. This makes MEME less sensitive to chance similarities than if a match threshold were set and all positions with z_{ij} value above that threshold were completely erased.

3. Experimental results

This section describes experiments using MEME that were conducted on two datasets. In all cases, the model used by MEME was the n-per model. The first dataset, which will be referred to as the CRP/LexA dataset, comprises DNA fragments which contain binding sites for the CRP and LexA regulatory proteins. The CRP/LexA dataset consists of all of the samples in the CRP dataset plus all the samples in the LexA dataset, which are described below. The second dataset, which will be referred to as the promoter dataset, contains samples of prokaryotic promoter regions. It is also described in detail below. An overview of the contents of the datasets is given in Table 1.

The CRP dataset is taken from Stormo and Hartzell (1989) who, in turn, derived it from Berg and von Hippel (1988) and de Crombrugghe *et al.* (Benoit et al., 1984). It contains 18 DNA fragments from *E. coli* each believed to contain one or more CRP binding sites. The dataset contains 18 CRP binding sites which had been verified by DNase protection experiments when the dataset was compiled. Some of the fragments contain putative CRP binding sites which have been determined by sequence similarity to known binding sites only. Each fragment in the dataset contains 105 bases and the fragments are not aligned with each other in any particular way.

The LexA dataset is taken from Table I in Hertz, et al. (1990). It contains 16 DNA fragments each believed to contain one or more LexA binding sites. The dataset contains 11 LexA binding sites which had been verified by DNase protection experiments when the dataset was compiled. An additional 11 putative LexA binding sites, as determined by sequence similarity to known binding sites, are also present in the dataset. Most of

the fragments contain 100 bases preceding and 99 bases following the transcription start position of a gene. Three of the fragments are shorter because 200 bases flanking the start position of the gene were not available. One of the samples in the LexA dataset overlaps a sample in the CRP dataset. The overlap includes the known CRP site.

The promoter dataset is taken from Cardon and Stormo (1992). It contains 231 *E. coli* DNA fragments each believed to contain promoter regions. This dataset was originally compiled by Harley and Reynolds (1987), and contained 288 fragments, but Cardon and Stormo omitted a number of fragments that were from highly redundant sequences or known to be mutant promoters. All the fragments roughly comprise positions -50 to $+10$ with respect to the start of transcription.[8] Previous work such as that of Harley and Reynolds (1987) has shown that the promoter motif seems to consist of two highly conserved sub-motifs of width 6 each, separated by a variable-length spacer. The spacer is usually 15, 16, 17 or 18 bases long. Although MEME cannot directly model such a variable-length motif, it can indirectly by discovering the two highly conserved ends of such motifs.

Table 1. Overview of the contents of the datasets.

dataset	*samples*	*average length of samples*	*proven CRP sites*	*proven LexA sites*
CRP	18	105	18	0
LexA	16	192	1	11
CRP/LexA	34	150	19	11
promoter	231	58	NA	NA

3.1. MEME can discover two different binding site motifs

MEME was run for 5 passes on the CRP/LexA dataset with $W = 20$, $NSITES = 17$. The value for W was chosen based on prior knowledge from the literature that this is the approximate size of both the CRP and LexA binding sites in DNA base-pairs.[9] The value for $NSITES$ was chosen arbitrarily as half the number of sequences in the dataset, because there are roughly that many footprinted sites of each type in the dataset. As mentioned previously, the exact value of $NSITES$ is not critical for MEME to discover the motifs. The first pass of MEME yielded an excellent model for the LexA binding site. The second pass produced a model for the CRP binding site. Subsequent passes produced models of unknown significance. The results of MEME on CRP/LexA are summarized in Table 2.

The model produced by the first pass of MEME on CRP/LexA identified and characterized the LexA binding site extremely well. The quality of the model can be judged partly from the degree to which it correctly identifies the known LexA binding sites in the dataset. One way of using the model produced by MEME is to examine the values of z_{ij} to see which positions in which samples in the dataset are given high probabilities of being the start of a motif. MEME prints the four highest values of z_{ij} for each sample in the dataset after each pass. Table 3 shows the values of z_{ij} after pass 1 of MEME

Table 2. The models found by each pass of MEME on the CRP/LexA dataset can be visually summarized by the consensus sequence derived from the ρ matrix by choosing the letter with the highest probability. The values of information content and log(*likelihood*) give a qualitative idea of the statistical significance of the model. Higher values imply the model is more significant. The models found for LexA and CRP on passes 1 and 2 of MEME have considerably higher log(*likelihood*) and information content than the models found on later passes. Note that $W = 20$ and $NSITES = 17$.

pass	*starting subsequence*	*final consensus*	I_{model}	log(*likelihood*)
1	TACTGTATATAAAACCAGTT	TACTGTATATATATACAGTA	13.206	-435.174
2	TTATTTGCACGGCGTCACAC	TTTTTTGATCGGTTTCACAC	9.087	-515.837
3	ATTATTATGTTGTTTATCAA	TTTATTTTGATGTTTATCAA	6.527	-539.083
4	TGCGTAAGGAGAAAATACCG	TGCGTAAGAAGTTAATACTG	7.912	-531.419
5	CAAATCTTGACATGCCATTT	CAAATATGGAAAGGCCATTT	8.027	-533.662

for the known LexA binding sites. It can be easily seen that the model found in the first pass characterizes the LexA binding site. Furthermore, all other values of z_{ij} were below 0.17, so the model appears to be very specific for the LexA binding site.

The consensus sequence for the model discovered in pass 1 of MEME on the CRP/LexA dataset also agrees exceedingly well with the LexA binding site. MEME prints the consensus (i.e., the most probable letter for each position in the motif as determined from ρ) after each pass. The consensus after pass 1 was TACTGTATATATATACAGTA, which matches the consensus reported by Hertz, et al. (1990) and is a perfect DNA palindrome.

Another way of seeing how well the model that was learned during pass 1 of MEME characterizes the LexA binding sites is to plot the information content score of each subsequence of the input data. Figure 1 shows the information content scores of both the CRP and LexA samples under the first pass model. (All scores below zero have been set to zero in the figure to make it easier to interpret.) It can easily be seen that the model gives the known binding sites high scores while most other subsequences receive low scores.

On the next pass, MEME discovers the CRP motif. The consensus sequence it reports for pass 2 is TTTTTTGATCGGTTTCACAC, which agrees well with the consensus found with one-per model and reported in Lawrence & Reilly (1990). More significantly, the model characterizes the CRP motif well, judging from the values of z_{ij} for the various positions in the samples in the dataset. Table 4 shows the values of z_{ij} found during pass 2 on the CRP/LexA dataset. According to (Lawrence & Reilly, 1990), the CRP dataset contains 24 known CRP binding sites, 18 of which had been verified by protection experiments. The value of z_{ij} for eight of these is above 0.99 in the model, while eleven have z_{ij} values above 0.1. It turns out that three of the samples from the LexA dataset also contain CRP binding sites. The sample labeled colicin E1 in the LexA dataset is actually from the same sequence and overlaps the sample labeled cole 1 in the CRP dataset. The overlap contains the CRP motif. LexA samples colicin Ia and colicin Ib also appear to contain CRP sites which are virtually identical to the colicin E1/cole 1 CRP site. For these sites z_{ij} is over 0.999, which is extremely high. Because of the overrepresentation

Table 3. Values of z_{ij} for the model found by MEME in pass 1 on the CRP/LexA dataset at the positions of the known LexA sites. Virtually all of the known sites have very high values of z_{ij} compared to the rest of the positions in the samples. The table shows the positions of the known sites (*site 1*, *site 2* and *site 3*) and the values of z_{ij} of the model at those positions. All other positions have values of z_{ij} below 0.17. Although the site at position 112 in the colicin E1 sequence has z_{ij} value only 0.05, this is one of the four highest z_{ij} values for this sequence. No proven sites are known for *himA* and *uvrC* and z_{ij} for all positions in those samples was very low, less than 0.0001.

sample	*site 1*	z_{ij}	*site 2*	z_{ij}	*site 3*	z_{ij}
cloacin DF13	97[a]	0.998684				
colicin E1	97	0.948441	112	0.051543		
colicin Ia	99[a]	0.998709				
colicin Ib	99[a]	0.990472				
recA	71	0.999987				
recN	71	0.999988	93	0.865704	111[a]	0.134281
sulA	85[a]	0.999990				
umuDC	91	0.999931				
uvrA	60	0.987786				
uvrB	71	0.999972				
uvrD	102	0.998539				
colicin A	34[a]	0.683563	48[a]	0.314723		
lexA	76	0.999982	55	0.999933		
mucAB	49[a]	0.999978				
himA						
uvrC						

[a] Indicates site known only by sequence similarity to known sites.

of this particular "version" of the CRP binding site, the model learned during pass 2 seems to be biased towards representing the version of the CRP binding site present in the colicin genes. This may explain why the model does not fit all of the CRP sites equally well.

Figure 2 shows the information content scores of the CRP/LexA dataset computed with the specificity matrix learned during pass 2 of MEME. Although the model is not as well defined as that of pass 1, it clearly matches the known CRP sites to a large degree.

3.2. *MEME can discover two parts of a single binding site*

MEME was run for 5 passes on the promoter dataset with $W = 6$, $NSITES = 231$. The value for W was chosen based on prior knowledge derived from the literature that this is the approximate size of both the -10 and -35 regions of *E. coli* promoters. The value of $NSITES$ was chosen based on the assumption that each sample in the dataset contains a promoter. The first pass of MEME yielded a model whose consensus was TATAAT, which is the known -10 region consensus. The second pass produced a model whose consensus was TTTACA, which is very close to the conventional -35 region consensus,

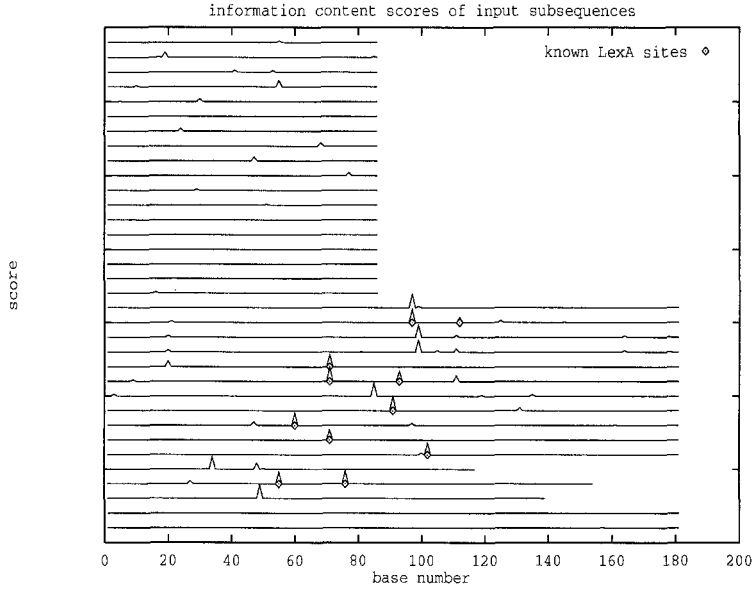

Figure 1. The information content score of each subsequence of the CRP/LexA dataset using the specificity matrix found on pass 1 of MEME. The CRP samples are the short curves at the top, while the LexA samples are the long curves at the bottom. Vertical scale is such that highest peak is 24.3 bits. All values below zero have been set to zero.

TTGACA. Further passes produced models of unknown significance. The results of MEME on the promoter dataset are summarized in Table 5.

The models learned on the first two passes of MEME on the promoter dataset are applied to the first thirty samples in the dataset and the information content score of each subsequence in the dataset is plotted in Figures 3 and 4. The base corresponding to the start of transcription of each sample is at position 50 on the horizontal axis of each plot. A column of peaks at position 37 in Figure 3 shows that the model identifies the -10 consensus region of the promoters. A column of peaks at position 15 of Figure 4 confirms that the second model identifies the -35 region of the promoters, even though its consensus sequence is slightly different from the generally accepted one.

4. Robustness of the MEME algorithm

The CRP/LexA dataset and the promoter dataset were also used to test the usefulness of the various separate ideas entering into the design of the MEME algorithm, and to evaluate the sensitivity of the algorithm to the particular values chosen for several parameters. Overall, the algorithm appears to be gratifyingly robust. Except where noted, MEME was run using the n-per model.

Table 4. Values of z_{ij} for the model found by MEME in pass 2 on the CRP/LexA dataset at the positions of the known CRP sites. Of 24 known CRP sites, eight have very high values of z_{ij}, and twelve more (those not stated as below some bound) have values of z_{ij} among the top four z_{ij} values for the given sequence. The three last three sites (labeled *b*, *c*, and *d*) are actually from the LexA dataset, not the CRP dataset. The sequence named colicin E1 actually is from the same gene as cole 1 and overlaps it in the CRP site region. The site in colicin Ia may not have been reported previously, and the colicin Ib site was previously reported as being a LexA site.

sample	*site 1*	z_{ij}	*site 2*	z_{ij}
cole1	17	$< .0004$	61	0.999185
ecoarabob	17	$< .0003$	55	0.999051
ecobglr1	76	0.028134		
ecocrp	63	0.998985		
ecocya	50	0.006001		
ecodeop	7[a]	0.999845	60	0.018088
ecogale	42	0.497545		
ecoilvbpr	39[a]	$< .0015$		
ecolac	9	0.996939	80	0.002302
ecomale	14[a]	0.997871		
ecomalk	29[a]	0.00129	61	0.035443
ecomalt	41	0.014568		
ecoompa	48	0.177722		
ecotnaa	71[a]	0.999222		
ecouxu1	17	0.998583		
pbr-p4	53	0.004511		
trn9cat	1	$< .0001$	84	0.000148
tdc	78[a]	0.506702		
colicin E1	27[b]	0.999186		
colicin Ia	13[c]	0.999692		
colicin Ib	13[d]	0.999333		

[a] Indicates site known only by sequence similarity to known sites.
[b] This LexA dataset sample overlaps CRP sample cole 1.
[c] This site may not have been reported previously.
[d] This apparent CRP site may have been confused with a LexA site by Varley and Boulnois (1984) and Hertz *et al.* (1990).

4.1. Subsequence-derived starting points work well with EM

The idea of running EM for only one iteration from starting points derived from each possible subsequence of the input dataset was tested. As the following experiments demonstrate, this method appears to work well at predicting good starting points from which to run EM to convergence. The experiments consisted of running EM for one iteration from each possible subsequence-derived starting point on the two datasets. The likelihood of each of the models thus obtained was plotted against the starting position of the subsequence from which the starting point was derived. Thus, one point was plotted for each position in each sample in the dataset. It was hoped that some starting points would yield models with significantly higher likelihood even after just one iteration.

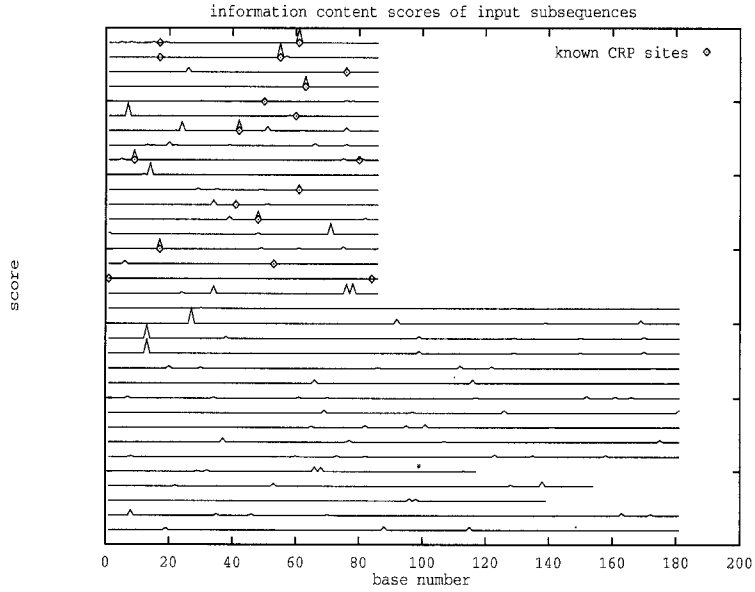

Figure 2. The information content score of each subsequence of the CRP/LexA dataset using the specificity matrix found on pass 2 of MEME. The CRP samples are the short curves at the top. The strong match of the model to three colicin samples in the LexA dataset is seen in the second, third, and fourth long curves. The vertical scale is such that highest peak is 18.92 bits. All values below zero have been set to zero.

Table 5. The models found on each pass of MEME on the promoter dataset are summarized as consensus sequences. The -10 and -35 region models were found on the first two passes of MEME and have much higher log(*likelihood*) and information content than the other models found.

pass	*starting subsequence*	*final consensus*	I_{model}	log(*likelihood*)
1	TAAAAT	TATAAT	4.627	-1409.458
2	TTTTTT	TTTACA	5.388	-1320.208
3	TGAAAA	TGAAAA	4.210	-1657.897
4	TATACT	TATACT	4.191	-1689.300
5	TTGCGC	TTGCGC	4.727	-1709.490

Then EM could be run to convergence from those starting points and the most likely model thus obtained could be selected as the output of MEME.

In the first experiment, the combined CRP/LexA dataset was used. The MEME algorithm was run with only one iteration of EM from each possible starting point. When the log(*likelihood*) values of the derived models are plotted against the position on the sequence from which the starting point was derived, it can be seen in Figure 5 that large peaks in the likelihood function were occurring in most of the LexA samples. (If the information content scores were plotted, the graph would have a very similar appearance. Since EM maximizes the likelihood of the model and not its information content, log

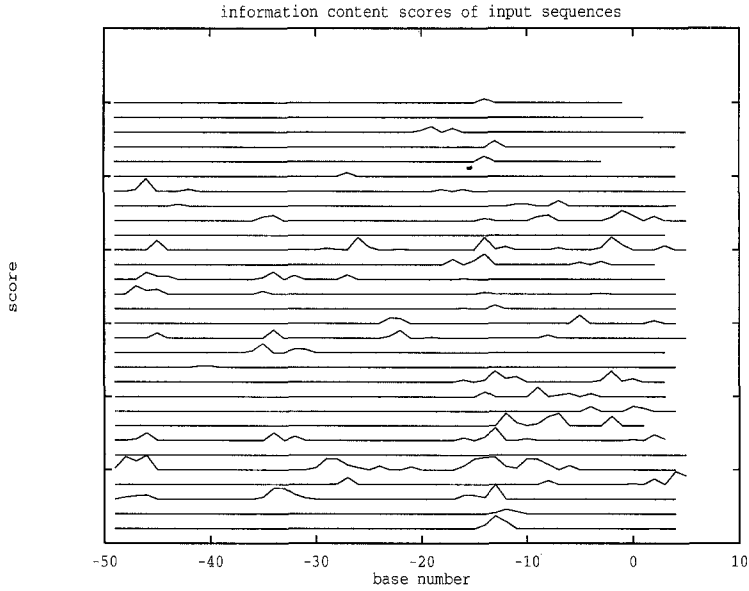

Figure 3. The information content score of each subsequence of the first 30 sequences of the promoter dataset using the specificity matrix of pass 1 of MEME. The concept learned on pass 1 of MEME on the promoter dataset locates the −10 region of the promoters. The vertical scale is such that highest peak is 7.21 bits. All values below zero have been set to zero.

likelihood was chosen as the criterion for choosing starting points. Information content could also be used, with similar results.)

Further investigation showed that the peaks tended to occur at the positions of the known LexA binding sites. Figure 6 shows an expanded view of the curve for the sample from recN. The recN sample contains three LexA binding sites whose left ends are marked on the horizontal axis of the figure. The peaks in the curve occur at or near these positions. The same phenomenon was observed for the other LexA samples, except for himA and uvrC which previous researchers (Hertz, et al., 1990) have noted do not match the LexA consensus

4.2. "Erasing" one motif is necessary to find another

On closer inspection of the plots, peaks could also be seen in the curves from the CRP samples at positions corresponding to known CRP binding sites. Figure 7 shows the expanded view for the CRP sample tnaa. As can be seen in the figure, it is difficult to distinguish the peaks generated by starting points derived from subsequences at the CRP binding sites from other peaks which do not correspond to any known sites. It appears that the other peaks are due to EM starting to converge to a model related to the LexA motif. Even a bad model of the highly conserved LexA motifs may have $\log(\textit{likelihood})$ similar to the best model of the CRP binding sites, due to the fact that

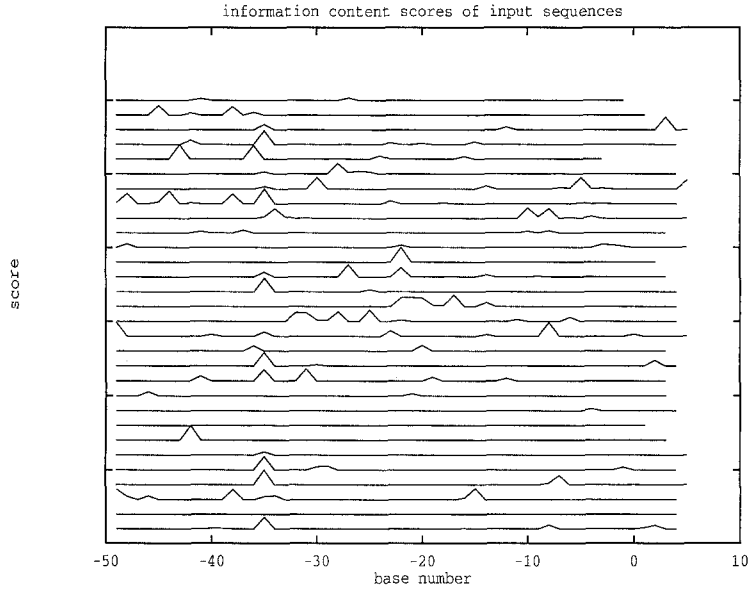

Figure 4. The information content score of each subsequence of the first 30 sequences of the promoter dataset using the specificity matrix of pass 2 of MEME. The concept learned on pass 2 of MEME on the promoter dataset locates the −35 region of the promoters. The vertical scale is such that highest peak is 7.74 bits. All values below zero have been set to zero.

the LexA binding sites are much more highly conserved than the CRP binding sites. The highest peaks produced by subsequences from the CRP samples were much lower than the highest peaks produced by the LexA samples. Also, no CRP sample produced a peak at a position corresponding to a CRP binding site that was clearly higher than all peaks produced from other subsequences of the CRP samples. This shows the necessity of somehow eliminating the LexA binding sites from the data in order to be able to discover the best starting points from which to run EM to learn a model for the CRP binding sites.

4.3. The expected number of motif appearances is not critical

If the choice of *NSITES* were critical to the ability of MEME using the n-per model to find one or more distinct motifs or parts of motifs in a dataset, it would be necessary to know in advance how many appearances of each motif were in the dataset. This would restrict the usefulness of MEME in discovering completely new motifs from sequence data alone. Fortunately, MEME discovers models for motifs with *NSITES* set to a wide range of values. So running MEME with just a few values of *NSITES* will probably suffice to find most motifs (if any) which are represented in a dataset.

MEME was run on the CRP/LexA dataset with various values of *NSITES* and all other parameters fixed. The models found by MEME on each pass were examined to see if they

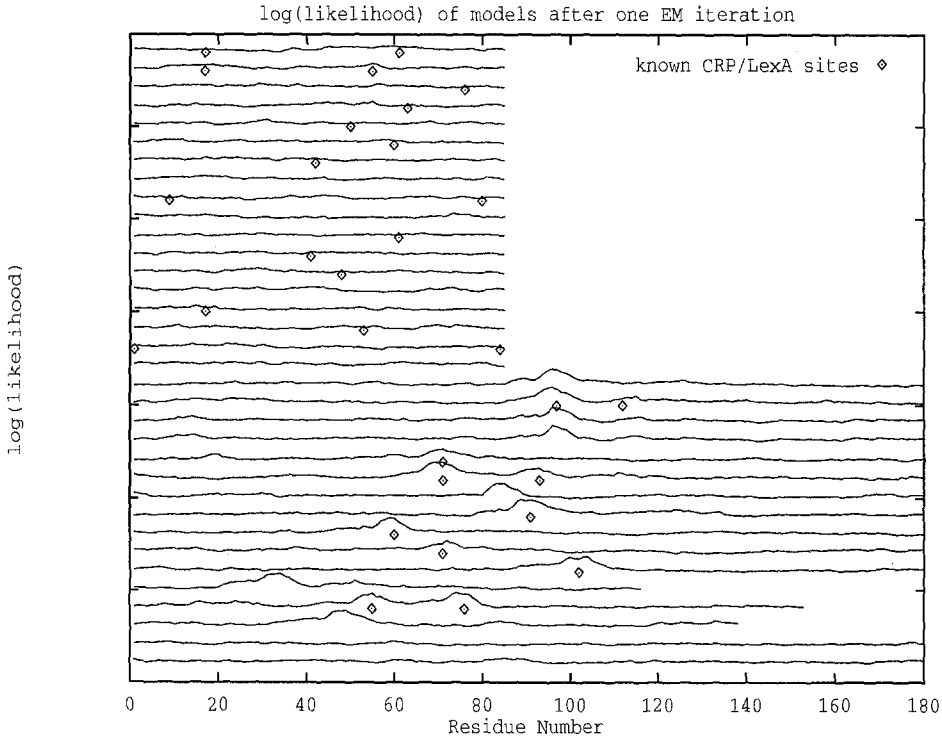

Figure 5. log(*likelihood*) after one iteration of EM from starting points derived from each possible subsequence in the CRP/LexA dataset. EM appears to converge quickly from starting points derived from subsequences at or near the LexA binding sites. The short curves at the top are the CRP samples, while the longer curves are the LexA samples. The vertical axis for each curve is scaled such that the highest peaks are at -481.6 and the lowest valleys are at -642.5.

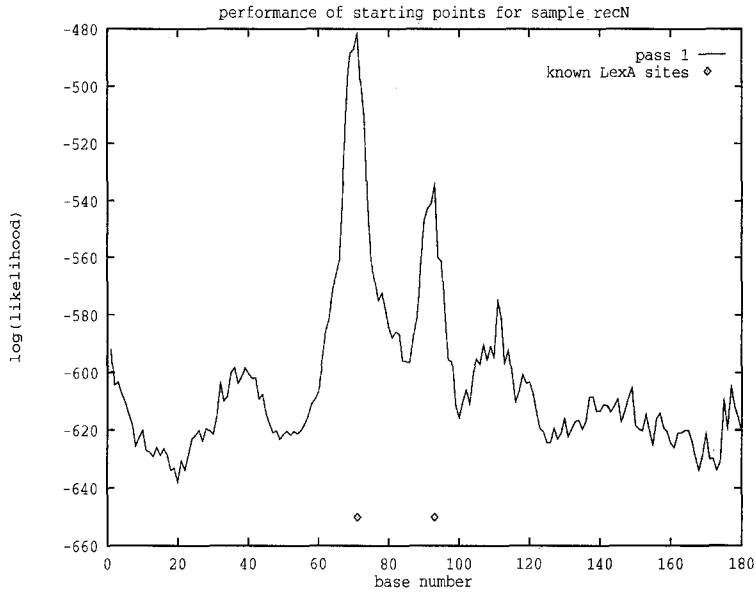

Figure 6. EM finds models of high likelihood when run for one iteration on the CRP/LexA dataset from starting points derived from subsequences of sample recN. The starting points correspond well with the known LexA binding sites, whose left ends are indicated on the horizontal axis.

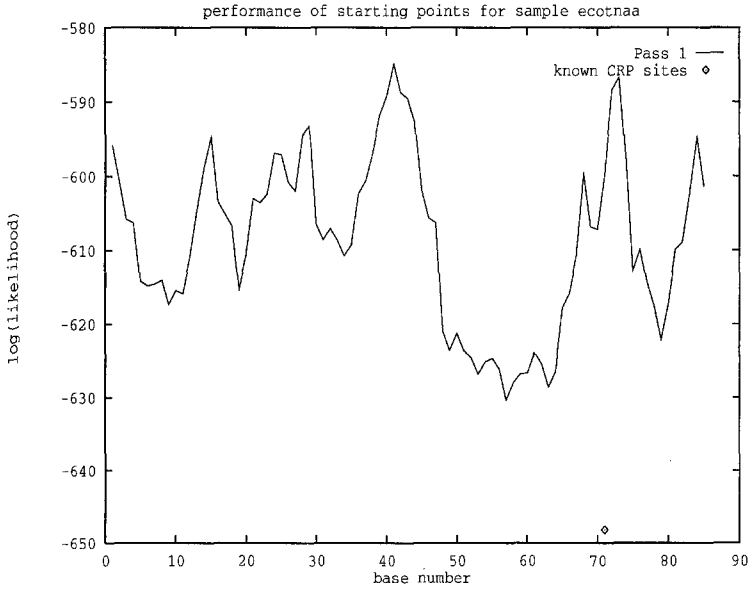

Figure 7. The log(*likelihood*) of the model after 1 iteration of EM in MEME varies strongly with the starting point. The plot shows the log(*likelihood*) of the model after one iteration of EM on dataset CRP/LexA run from the starting points generated from the subsequences in the sample labeled "tnaa".

fit the known consensus sequences for LexA and CRP. Table 6 shows the passes of MEME on which models for LexA and CRP motifs were discovered and the information content and log(*likelihood*) of the models. MEME always finds a model for the LexA motif on the first pass. With low *NSITES*, it finds LexA more than once, due presumably to the fact the LexA binding sites do not get completely erased. (MEME effectively erases at most *NSITES* occurrences of a motif after each pass, so if $NSITES = 5$ and there are fifteen LexA binding sites, there are still enough left for pass 2 to find another model of the LexA motif.) MEME found a model of the CRP motif within four passes for all values of *NSITES* tried except for $NSITES = 5$. Usually, CRP was the second model found. While the values of information content and log(*likelihood*) of the LexA models were always much higher than those of all other models found by MEME, this was not always true for the CRP models. Only when *NSITES* was close to the actual number of known CRP binding sites in the dataset was the information content and log(*likelihood*) of the CRP model much higher than for the other models (of unknown biological significance) found by MEME.

4.4. The n-per model is less sensitive to noise than the one-per model

The removal of the one-motif-appearance-per-sequence assumption was intended, among other things, to make the n-per model less sensitive to noise than one-per model. For example, if it is suspected that one or more of the sequences in a dataset is noise (i.e., does not contain an appearance of a motif), *NSITES* can be set to a value which is less than than the number of sequences in the dataset. If MEME correctly locates just the appearances of the motif, the model found will have higher log(*likelihood*) than that found by using the one-per model which is forced to choose an appearance in every sequence in the dataset. To test this assumption, MEME was run with both the one-per model and the n-per model on datasets which contained varying numbers of randomly generated sequences (with *NSITES* set to the same, fixed value each time). The random sequences had the same letter frequencies as the dataset as a whole, and they were the same length. The datasets used were CRP and LexA with various numbers of random sequences added. In both cases, MEME with the n-per model learned the correct concept on the first pass from datasets with more random sequences than the MEME using the one-per model could tolerate. MEME with the n-per model learned a model for the CRP binding site with 30 random sequences added to the 18 sequences of the CRP dataset. (It learned the model even with 50 random sequences, although then it learned it on the second pass.) MEME with the one-per model was not able to learn a LexA binding site model with more than 60 random samples added to the dataset, and it learned an "off-center" model when more than 20 random samples were in the dataset. MEME with the n-per model, however, learned the correct LexA model even with 80 random samples added to the dataset.

Figure 8 shows the information content of the CRP and LexA models learned by MEME with the n-per model and the one-per model on the first pass from datasets with various numbers of random sequences added. The CRP models learned with the n-per model also consistently had higher information content than those learned with the one-per model.

Table 6. MEME finds models of the LexA and CRP binding sites when *NSITES* has values between 10 and 35. When *NSITES* is above 10, LexA and CRP are usually found on the first two passes. Only with *NSITES* = 5 did MEME fail to find CRP on any of the first five passes.

NSITES	*pass*	*consensus*	I_{model}	log(*likelihood*)	*motif*
5	1	ATACTGTATATAAAAACAGT	8.151	-154.337	LexA
	2	AATACTGTATATGTATCCAG	7.667	-158.139	LexA
	3	TGTGAAAGACTGTTTTTTTG	6.968	-161.024	?
	4	ACTATCATCAAATCTTGACA	5.406	-169.906	?
	5	GATGCGTAAGCAGTTAATTC	6.280	-167.133	?
10	1	TACTGTATATAAAAACAGTA	11.740	-271.797	LexA
	2	TAATACTGTATATGTATCCA	7.318	-319.596	LexA
	3	GTGAAAGACTATTTTTTTGA	8.460	-303.710	?
	4	TTTCTGAACGGTATCACAGC	8.145	-317.833	CRP
	5	AAGCAGATTATGCTGTTGAT	6.895	-318.830	?
15	1	TACTGTATATATATACAGTT	13.513	-379.939	LexA
	2	TTTTTTGAACGATTTCACAT	9.198	-454.496	CRP
	3	TTTATTTTGATGTTTATCAA	6.620	-475.009	?
	4	TGCGTAAGAAGTTAATACTG	7.947	-471.933	?
	5	CAAAAATGGAAAGCCATTTT	7.292	-481.090	?
20	1	AATACTGTATATATATACAG	12.883	-520.728	LexA
	2	TTTTTGAACGGTTTAAAATT	8.237	-603.571	CRP
	3	ATTATTGTGATGTTGATTAT	7.075	-634.142	?
	4	TGCGGAAGCAGATAATACTG	8.042	-627.719	?
	5	ATGAAAGTCTACATTTTTGT	7.042	-638.444	?
25	1	TACTGTATATATATACAGTA	12.161	-669.214	LexA
	2	TTTATTTTGATGTTTTTCAA	7.797	-760.468	?
	3	TTTCTGAAAGGTATAACATC	7.739	-786.765	CRP
	4	CAAAAATGGAAAAGCAATTT	7.676	-789.667	?
	5	TGCGTAAGAAGATAATACTG	7.253	-803.956	?
30	1	TACTGTATATATATACAGTA	11.087	-828.649	LexA
	2	TTTTTGTGATCTGTATCACA	7.842	-929.059	CRP
	3	CAAAAATGGATAACCATTTT	7.529	-952.776	?
	4	TATGCGTAAGCAGTAAAATT	7.401	-953.792	?
	5	TGAGGATGATAACGAATATC	6.820	-975.923	?
35	1	TACTGTATATATATACAGTA	10.300	-995.800	LexA
	2	ATTATTGTGATGTTGATCAT	7.247	-1092.196	CRP
	3	CAAAAATGGAAAACCATTTT	7.425	-1112.207	?
	4	TTTCTGACCCAGTTCACATT	7.717	-1104.486	CRP
	5	ATGCGTAAGCAATTTATTCA	6.826	-1135.477	?

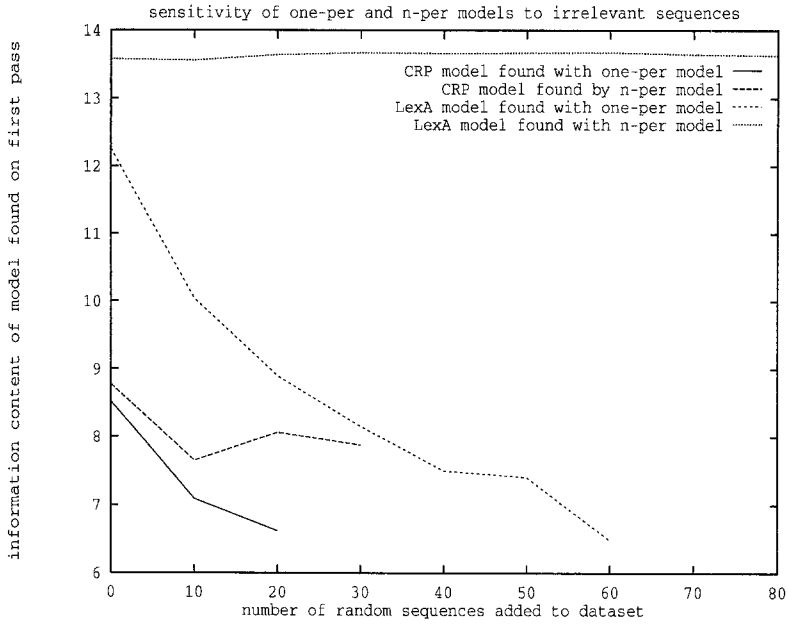

Figure 8. The information content of the LexA and CRP models found on the first pass of MEME with the n-per model and the one-per model, run separately on the CRP and LexA datasets with different numbers of random examples added. The comparative advantage of the n-per model is clear. Especially with motifs whose occurrences are highly conserved, the n-per model finds very good models even when many sequences not containing the motif are present. MEME was run with $W = 20$ and $NPASSES = 1$. $NSITES$ was set to 15 for the n-per model.

This was true even for the model learned with no random sequences added to the dataset. Presumably, this is indicative of the fact that the n-per model is taking advantage of the sequences with multiple appearances of the CRP site. The models learned with the n-per model for LexA were extremely robust to the number of random samples added to the dataset. There was almost no decrease in the information content no matter how many random samples were present. The one-per model, on the other hand, found models with lower information content when more random samples were in the dataset.

It is clear from Figure 8 that MEME using the n-per model will find a set of highly conserved binding sites even in datasets where the vast majority of the sequences do not contain it. The one-per model suffers from the fact that it must always average in one supposed motif appearance from each sample. MEME with the n-per model is thus able to deal with a particular type of noise—samples containing no motif appearances—if a good estimate of the true number of motif appearances ($NSITES$) is available.

5. Discussion

The MEME algorithm demonstrates the power of several new ideas. Subsequence-derived starting points have been shown to be a powerful way of selecting starting points for EM, and may be useful with other methods as well. Since EM tends to converge quickly from good starting points, MEME saves a great deal of time by only running EM for one iteration from each starting point and greedily selecting the best starting point based on the likelihood of the learned model. The modifications to the EM algorithm which allow MEME to drop the assumption that each sequence contains exactly one appearance of a motif and fit the n-per model to a dataset have been shown to give MEME the ability to discover motifs in datasets which contain many sequences which do not contain the motif. Finally, the probabilistic weighting scheme used by MEME to erase appearances of the motif found after each pass was demonstrated to work well at finding multiple different motifs as well as motifs with multiple parts.

The MEME algorithm should prove useful in analyzing biological sequence data. It is a robust tool for discovering new motifs from sequence data alone when little or no prior knowledge is available. When MEME is used to discover motifs from sequence data alone, it is performing unsupervised learning. Effectively, MEME finds clusters of similar subsequences in a set of sequences. Some measure of the unlikeliness of a cluster, information content of the model for example, can then be used to decide if other methods (i.e., wetlab experimentation) should be applied to verify that the sites which match the model actually are biologically related. Plots of information content scores of various positions of the sequences in the dataset such as in Figure 1 and Figure 2 can also be helpful to a biologist for discovering which clusters are significant and which may be statistical artifacts.

When MEME is used with a dataset of sequences each of which is known to contain a motif, such as the promoter dataset, it is performing supervised learning. Because the models MEME learns do not allow a motif to have variable length (i.e., no insertions or deletions are allowed), MEME is limited to learning a restricted class of motifs. It may be possible to use the multiple models learned by MEME on passes through the dataset as features for another learning algorithm. For example, a decision tree learner such as ID3 (Quinlan, 1986) or CART (Breiman et al, 1984) could use the models learned by MEME on the promoter dataset as features to learn a classification rule for *E. coli* promoters. Since the first two passes of MEME found models for the -10 and -35 regions of the promoter, this approach should have a high chance of success. Another promising idea is to use the short motifs learned by MEME to construct starting points for hidden Markov models.

The innovations added to the EM algorithm in MEME can also be used with hidden Markov models (HMMs) (Haussler, et al., 1993). The idea of using subsequence-derived starting points may be adaptable for use with HMMs. The method used by MEME for probabilistically erasing sites after each pass would certainly be easy to add to the standard forward/backward HMM learning algorithm. It should also be possible to design a HMM which, like the n-per model, eliminates the assumption of one motif per sequence. It may

also be possible to adapt MEME innovations to learning stochastic context free grammars for biopolymer concepts (Sakakibara, et al., 1993).

MEME discovered CRP sites in the colicin Ia and colicin Ib samples. The site in colicin Ib was mentioned in Varley and Boulnois (1984) as being either a LexA site or possibly a CRP site. Hertz, et al. (1990) appear to have classified it as a LexA site. The results reported here indicate that the site is probably a CRP binding site, not a LexA binding site: the information content score for the site under the CRP model was around 16, whereas it was less than 1 under the LexA model. No mention of the CRP site found in colicin Ia was found in the literature.

Acknowledgments

This work was supported in part by the National Science Foundation under Award No. IRI-9110813, and Timothy Bailey is supported by an NIH Genome Analysis Pre-Doctoral Training Grant No. HG00005. The authors are grateful to Michael Gribskov for many useful conversations during the course of the work reported here, to Douglas W. Smith for extensive suggestions during the writing of this article, and to several other colleagues for advice and encouragement.

Appendix

Reestimating ρ and z for the one-per and n-per models.

During each iteration of EM, the values of the letter probabilities of the motif model ρ, and of the offset probabilities z, must be reestimated. With the one-per model, the z values are reestimated using Bayes' rule from the current estimate of ρ. For both models, given the values of z, ρ is estimated as the expected values of the letter frequencies. How this is done is described below.

To describe the EM algorithm for the two model types formally, the following definitions are useful. Let N be the number of sequences, W be the length of the motif, and L be the length of each sequence (assume all are of the same length). Define $z_{ij}^{(q)}$ as the estimate after q iterations of EM of the probability that the site begins at position j in sequence i given the model and the data. Let $\rho_{lk}^{(q)}$ be the estimate after q iterations of EM of the probability of letter l appearing in position k of the motif. Let S_i be the ith sequence in the dataset and S_{ij} be the letter appearing in position j of that sequence. Define an indicator variable Y_{ij} that equals 1 if the site starts at position j in sequence i, and 0 otherwise.

We ignore the probability of the letters outside of the motif, and only consider the probability of the letters in the motif. For both model types, EM must calculate the probability of sequence S_i given the motif start and the model. This can be written as

$$P(S_i|Y_{ij}=1,\rho^{(q)}) = \prod_{k=1}^{W} \rho_{l_k,k}^{(q)}$$

where the sequence S_i has letter l_k at position $j+k-1$, i.e., $S_{i,j+k-1} = l_k$. This forms the basis for calculating $z^{(q)}$.

With the one-per model, Bayes' rule is used to estimate $z^{(q)}$ from $P(S_i|Y_{ij} = 1, \rho^{(q)})$. Bayes' rule states that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

so

$$z_{ij}^{(q)} = P(Y_{ij} = 1 \mid \rho^{(q)}, S_i) = \frac{P(S_i|Y_{ij} = 1, \rho^{(q)})P^0(Y_{ij} = 1)}{\sum_{k=1}^{L-W+1} P(S_i|Y_{ik} = 1, \rho^{(q)})P^0(Y_{ik} = 1)}$$

where $P^0(Y_{ij} = 1)$ is the prior probability that the motif begins at position j in sequence i. P^0 is not estimated and is assumed to be uniform,

$$P^0(Y_{ij} = 1) = 1/(L - W + 1),\ k = 1, \ldots, (L - W + 1)$$

so the above simplifies to

$$z_{ij}^{(q)} = \frac{P(S_i|Y_{ij} = 1, \rho^{(q)})}{\sum_{k=1}^{L-W+1} P(S_i|Y_{ik} = 1, \rho^{(q)})}$$

The probability is only estimated for sites which are completely within a sequence, so j is assumed to be within the range $1, \ldots, L - W + 1$ in all calculations of $z^{(q)}$.

Notice that the above formula for $z^{(q)}$ ensures that it sums to 1.0 for each sequence. This enforces the implicit assumption of the one-per model that each sequence contains exactly one appearance of the shared motif. For the n-per model, our modified EM algorithm normalizes $z^{(q)}$ so that the sum over all positions in all sequences is *NSITES*. This can be written formally as

$$z_{ij}^{(q)} = NSITES \frac{P(S_i|Y_{ij} = 1, \rho^{(q)})}{\sum_{n=1}^{N} \sum_{k=1}^{L-W+1} P(S_n|Y_{ik} = 1, \rho^{(q)})}$$

Once z has been calculated as above for the n-per model, it undergoes two normalizations to enforce the constraints that each $z_{ij}^{(q)}$ is less than or equal to 1.0, and that the sum of the $z_{ij}^{(q)}$ in any window of length W is less than or equal to 1.0. These constraints can be written formally as

$$z_{ij}^{(q)} \leq 1.0, \text{ for } 1 \leq i \leq N \text{ and } 1 \leq j \leq L$$

$$\sum_{j=k}^{k+W-1} z_{ij}^{(q)} \leq 1.0, \text{ for } 1 \leq i \leq N \text{ and } 1 \leq k \leq L - W + 1.$$

There are many different ways in which the constraints could be enforced. A particular manner was chosen which reduces computational effort. No claim is made that this is the only or best choice. The two constraints are enforced separately by applying the

following two algorithms in order. Figure A.1 presents the first algorithm, which makes one or more passes through the offset probabilities normalizing them to sum to *NSITES* and "squashing" (setting to 1.0) any that would exceed 1.0 after normalization. After each pass, if any offset probabilities get squashed, another pass is made to raise the value of offset probabilities that have never been squashed so that the *NSITES* total is enforced. In practice, usually few passes are needed. The second algorithm, given in Figure A.2, is run next to enforce the constraint that no window of W positions has offset probabilities that sum to more than 1.0. This is achieved by dividing each sequence into adjacent windows of length W and normalizing within each window separately. Windows are then shifted one to the right and the process is repeated. This is done for all W possible shifts of the windows, which guarantees that no window of width W will have offset probabilities summing to greater than 1.0, but may reduce the total sum below *NSITES*. The squashing algorithm could be repeated to correct this but this is not done in the interest of saving computation time.

Notes

1. The name MEME has several explanations. First, it is an acronym for multiple EM for motif elicitation. Second, as an English word "meme" means a theme or motif whose propagation through cultural evolution is similar to the propagation of a gene in biological evolution. Third, MEME is a greedy algorithm—a "me! me!" algorithm.
2. A related measure used occasionally in this paper, I_{model} is the information content of the model (Stormo, 1988). It is the sum of the information content of each position in the motif, I_j, over all the positions in the motif. The information content of a position in the motif is defined as

$$I_j = \sum_{l \in \mathcal{L}} \rho_{lj} \log(\frac{\rho_{lj}}{\mu_l}),$$

   where μ_l is the overall frequency of letter l in the dataset. The information content of the model is thus defined as

$$I_{model} = \sum_{j=1}^{W} I_j.$$

   The relationship between I_{model} and $\log(\textit{likelihood})$ is discussed by Stormo (1990) and Bailey (1993).
3. Promoter sequences are DNA sequences that precede genes and are necessary for the transcription of DNA to messenger RNA.
4. The consensus sequence of a motif is the sequence consisting of the most commonly occurring letter in each position of the appearances of the motif. Ties are resolved arbitrarily.
5. See Stormo (1988) for a discussion of matrix-based scoring of sequences.
6. Using all possible subsequences of the first dataset sequence is suggested in Stormo and Hartzell (1989). The MEME approach of using all subsequences of all sequences is preferable since it makes the order in which sequences are given unimportant. Not using just the first sample also eliminates the problem of the first sample happening to contain no motif occurrence.
7. The idea of a "conserved motif" comes from the biological idea that the occurrences of motifs are often related to each other by evolution. A well conserved motif is one whose appearances are all almost identical to each other because little mutation has occurred in them since they separated from each other or from a common ancestor.

```
1.  SQUASH ( z (unnormalized;  z_ij^(q) = P(S_i|Y_ij = 1, ρ^(q))),
2.      total (the total of z^(q) for all sequences and positions),
3.      NSITES (the number of appearances of the motif expected),
4.      L (length of the sequences),
5.      N (number of sequences)) {
6.      renormalize = true
7.      while (renormalize) {
8.          renormalize = false
9.          normalize = total/NSITES
10.         total = 0
11.         for i = 1 to N {
12.             for j = 1 to L − W + 1 {
13.                 p = z_ij
14.                 if (p < 1) {
15.                     p = p/normalize
16.                     if (p > 1) {
17.                         p = 1
18.                         NSITES = NSITES − 1
19.                         renormalize = true
20.                     }
21.                 }
22.                 z_ij = p
23.                 if (p < 1) total = total + p
24.             }
25.         }
26.     }
27.     return
28. }
```

Figure A.1. SQUASH: Normalize the z_{ij} to sum to $NSITES$ while constraining each to be between 0 and 1.

```
1.  SMOOTH ( z^(q) (normalized offset probabilities),
2.      L (length of the sequences),
3.      N (number of sequences) ) {
4.      for i = 1 to N {
5.          for offset = 1 to W {
6.              for j = offset to L − 2 * W by W {
7.                  localp = 0
8.                  for k = 1 to W {
9.                      localp = localp + z^(q)_{i,j+k}
10.                 }
11.                 if (localp > 1) {
12.                     for k = 1 to W {
13.                         z^(q)_{i,j+k} = z^(q)_{i,j+k} / localp
14.                     }
15.                 }
16.             }
17.         }
18.     }
19.     return
20. }
```

Figure A.2. SMOOTH: Constrain the sum of offset probabilities in any window of width W to sum to no more than 1.0.

8. Biologists often number the "bases" (i.e., letters) in a DNA sequence with base 1 being the base where transcription from DNA to messenger RNA begins. Bases preceding the start of transcription are given negative numbers, starting at -1, with 0 not used.)
9. If the best value of W is not known in advance, MEME can be run repeatedly with different values. Lawrence and Reilly (1990) addresses the question of choosing the best value of W. Each run of MEME uses a single value of W for all motifs found.

References

Bailey, T.L. (1993). Likelihood vs. information in aligning biopolymer sequences. Technical Report CS93-318, University of California, San Diego.

Bairoch, A. (1993). The PROSITE dictionary of sites and patterns in proteins, its current status. *Nucleic Acids Research*, 21(13):3097–3103.

Berg, O.G. & von Hippel, P.H. (1988). Selection of DNA binding sites by regulatory proteins. *Journal of Molecular Biology*, 200:709–723.

Breiman, L., Friedman, J. H., Olshen, R. A.& Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, California.

Cardon, L.R. & Stormo, G.D. (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *Journal of Molecular Biology*, 223:159–170.

de Crombrugghe, B., Busby, S. & Buc, H. (1984). Cyclic AMP receptor protein: Role in transcription activation. *Science*, 224:831–838.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

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

Harley, C.B. & Reynolds, R.P. (1987). Analysis of *E. coli* promoter sequences. *Nucleic Acids Research*, 15:2343–2361.

Haussler, D., Krogh, A., Mian, I.S. & Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 1, pages 792–802, Los Alamitos, CA. IEEE Computer Society Press.

Hertz, G.Z. Hartzell, III, G.W. & 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.

Kolakowski, L.F., Leunissen, J.A. &. Smith, J.E. (1992). ProSearch: fast searching of protein sequences with regular expression patterns related to protein structure and function. *Biotechniques*, 13:919–921.

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

Lawrence, C.E. & 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.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

Sakakibara, Yasubumi, Brown, Michael, Underwood, Rebecca C., Mian, I. Saira & Haussler, D. (1993). Stochastic context-free grammars for modeling RNA. Technical Report 93-16, UCSC-CRL.

Stormo, G.D. (1988). Computer methods for analyzing sequence recognition of nucleic acids. *Annual Review of Biophysics and Biophysical Chemistry*, 17:241–263.

Stormo, G.D. (1990). Consensus patterns in DNA. *Methods in Enzymology*, 183:211–221.

Stormo, G.D. & Hartzell, III, G.W. (1989). A tool for multiple sequence alignment. *Proceedings National Academy Science USA*, 86:1183–1187.

Uberbacher, E.C. & Mural, R.J. (1991). Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proceedings National Academy Science USA*, 88:11261–11265.

Varley, J.M. & Boulnois, G.J. (1984). Analysis of a cloned colicin Ib gene: complete nucleotide sequence and implications for regulation of expression. *Nucleic Acids Research*, 12:6727–6739.

Received September 30, 1993
Accepted July 27, 1994
Final Manuscript August 11, 1994