UNSUPERVISED LEARNING OF MULTIPLE MOTIFS IN BIOPOLYMERS USING EXPECTATION MAXIMIZATION

TIMOTHY L. BAILEY   (TBAILEY@CS.UCSD.EDU)

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

CHARLES ELKAN   (ELKAN@CS.UCSD.EDU)

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

ABSTRACT: The MEME algorithm extends the expectation maximization (EM) method 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 enhancements expand the range of problems which can be addressed by the EM algorithm and increase the probability of finding solutions. 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. 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. 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 pattern, motif, biopolymer, promoter, binding site

Contents

1   Introduction

The problem addressed by this report is that of identifying and characterizing shared motifs in a set of unaligned genetic or protein sequences. A motif is defined here as a family of nucleic or amino acid subsequences which each have some biological property of interest such as being DNA binding sites for a regulatory protein. In this report we are concerned only with *contiguous* motifs. That is, appearances of a motif may differ in point mutations, but insertions or deletions are not allowed. 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.

1.1   The expectation maximization (EM) method

Lawrence and Reilly [1990] introduced the expectation maximization (EM) method as a means of solving the simpler version of the problem mentioned above. The EM method takes as input a set of unaligned sequences and a motif length (*LSITE*) and returns a probabilistic model of the shared motif. The idea behind the method is that each sequence in the dataset contains an example of the motif. It is not known where the motif appears (what its starting offset is) in each example, however. If this were known, subsequences of length *LSITE* 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, multinomial 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 $poff_{ij}$ 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. The $poff_{ij}$

- Input:

  dataset of sequences

  LSITE (length of shared motif)

- Algorithm:

  choose starting point (*freq*)

  do {

     reestimate *poff* from *freq* with Bayes' rule

     reestimate *freq* from *poff*

  } **until** (change in *freq* < ϵ)

Figure 1: The basic EM algorithm.

are then used to reestimate the frequency of letter l in column c of the motif, $freq_{lc}$, for each letter in the alphabet and $1 \leq c \leq LSITE$. How the reestimations are done is described in Appendix A. The EM algorithm alternately reestimates *poff* and *freq* until *freq* changes very little from iteration to iteration. The notation *poff* is used to refer to the *matrix* of offset probabilities $poff_{ij}$. Likewise, *freq* refers to the *matrix* of letter frequencies $freq_{ij}$.

A pseudo-code description of Lawrence and Reilly's EM algorithm is given in Figure 1. EM starts from an estimate of *freq* provided by the user or generated at random.

The EM algorithm simultaneously discovers a model of the motif (the sequence of independent multinomial random variables with parameters *freq*) and estimates the probability of each possible starting point of examples of the motif in the sequences in the dataset (*poff*). The likelihood of the model given the training data is just the probability of the data given the model. The logarithm of this quantity is

$$\log(likelihood) = N \sum_{j=1}^{J} \sum_{l \in \mathcal{L}} freq_{lj} \log(freq_{lj}) + N(L-J) \sum_{l \in \mathcal{L}} fout_l \log(fout_l)$$

where N is the number of sequences in the dataset, L is the length of the sequences, $J = LSITE$ is the length of the shared motif, $\mathcal{L}$ is the alphabet of the sequences, and

$fout_l$ is the frequency of the letter l in all positions of the sequences outside the instances of the shared motif. 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.[1]

1.2   Limitations of the original EM method

The original version of EM suffers from several limitations. First, it is not clear how to choose a starting point (an initial value of *freq*) nor when to quit trying different starting points. This makes it difficult to be satisfied that the correct shared motif has been found. Second, EM 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 to find the shared motif at all. Finally, EM 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 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 on EM would make the algorithm 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, adds several extensions to EM to

---

[1]A related measure used occasionally in this paper is the relative entropy of the model, which is

$$RE = \sum_{j=1}^{J} \sum_{l \in \mathcal{L}} freq_{lj} \log(\frac{freq_{lj}}{null_l})$$

where $null_l$ is the overall frequency of letter l in the dataset. See [Bailey, 1993] for an explanation of the connection between RE and $\log(likelihood)$ as defined above.

overcome the limitations described above.[2] MEME chooses starting points systematically, based on all subsequences of sequences in the training dataset. It eliminates the assumption of one sequence/one motif and allows each sequence to contain zero, one or several appearances of the shared motif. Finally, MEME probabilistically "erases" the appearances of a site after it is found, and continues searching for further shared motifs in the dataset.

The MEME algorithm was run on two datasets. The first was a dataset combining 18 *E. coli* sequences containing CRP binding sites [Lawrence and 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 and Reynolds, 1987]. MEME discovered the `TATAAT` and `TTGACA` consensus sequences 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 method [Lawrence and Reilly, 1990]. Refer to Figure 2 for a pseudo-code description of the MEME algorithm. In the inner loop, the EM algorithm is run repeatedly with different starting points. The starting points are derived from actual subsequences which occur in the input dataset. EM is run only a few iterations, 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.

---

[2]The name MEME has several explanations. First, it is an acronym for <u>m</u>ultiple <u>EM</u> for <u>m</u>otif <u>e</u>licitation. Second, as an English word "meme" means a theme or motif whose propagation through cultural evolution is similar to the propagation of genes in biological evolution. Third, MEME is a greedy algorithm—a "me! me!" algorithm.

- Input:

    dataset of sequences

    $LSITE$ (length of shared motifs)

    $PASSES$ (number of distinct shared motifs to find)

    $NITER$ (number of iterations to run EM)

    $MAXP$ (number appearances of each shared motif expected in dataset)

- Algorithm:

    for motif = 1 to $PASSES$ {

      for each subsequence in dataset {

        run modified EM for $NITER$ iterations with

          starting point derived from this subsequence

      }

      choose model of shared motif with highest likelihood

      run EM to convergence from starting point which generated that model

      print converged model of shared motif

      "erase" appearances of shared motif from dataset

    }

Figure 2: The MEME algorithm.

2.1 Using subsequences as starting points for EM

Given different starting points (i.e. initial letter frequency matrices *freq*) the EM algorithm may converge to different final models. These models are at local maxima of the likelihood function described earlier. The correct model for the shared motif is probably given by the model which globally maximizes the likelihood function, but EM is not guaranteed to find the global maximum, on a local maximum. Previous authors [Lawrence and Reilly, 1990; Cardon and 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 the model for motifs we are discussing does 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 *LSITE* into a letter frequency 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 frequency matrix (i.e. model), EM should tend to converge to the global optima when run with them as starting points.[3]

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

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

---

[3]Using all possible subsequences of the first dataset sequence is suggested in [Stormo and Hartzell, III, 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.

and the consensus sequence is `TATA`. 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 frequency 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 four in this example) of the sequences in the dataset, it is reasonable to assume that at least one of them will be "close" to `TATA` 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 frequency matrix. One cannot simply convert it to a matrix with frequency 1.0 for the letter in the subsequence and 0.0 for all others, i.e. convert `TATA` to

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

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 `TATA` is

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

It would be highly expensive computationally to run EM until convergence from every possible starting point corresponding to some subsequence of length *LSITE* 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 or two iterations 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 (usually one or two) 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

The original EM method assumes that each sequence in the dataset contains exactly one appearance of the shared motif to be characterized. This assumption is implicit in 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 gets around this problem in a simple way. 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 $MAXP$, 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 $MAXP$ to be the expected number of appearances of the shared motif in the dataset. If $MAXP$ is set equal to the number of sequences in the dataset, it is possible for MEME to get approximately the same results as EM 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, MEME is able to choose models which assign the offset probabilities in any fashion which satisfies the two constraints mentioned above.

The relaxation of the one motif appearance per sequence constraint 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 $MAXP$ 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 original EM algorithm must assign offset probabilities summing to 1.0 to each sequence in the dataset. The effect of various settings for $MAXP$ is discussed in section 4.3. The exact value chosen for $MAXP$ is not critical and it is not necessary to know in advance 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 as though they were multiple appearances of a shorter sequence. For example, if $LSITE$ is 6, the sequence `AAAAAAAA` is treated by MEME roughly as though it were three appearances of the sequence `AAAAAA`. This is so because MEME might allow offsets 1, 2 and 3 of the sequence to have the maximum probability of 1.0. (The original EM algorithm would not allow this, since the total offset probability for a single sequence must sum to 1.0.) To overcome this difficulty, MEME enforces an additional constraint when calculating the offset probabilities. It renormalizes the offset probabilities so that no $LSITE$ adjacent offsets have probabilities that sum to greater than 1.0. This essentially makes MEME treat sequences like `AAAAAAAA` the same way as EM 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, the EM method may have trouble finding more than one of them. If the motifs have some similarity, the EM algorithm may always converge to the most conserved motif. Another possibility is the that the EM algorithm 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 $poff_{ij}$ 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 site is the product of $(1 - poff_{ij})$ for all $k - LSITE < j < k + LSITE$. 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 $poff_{ij}$, the weighted offset probabilities $w_{ij} \cdot poff_{ij}$ 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 $w_{i,(j+LSITE-1)}$ will all be less than $1 - poff_{ij}$. Hence they cannot contribute much to the reestimation of *freq* and are effectively erased. Notice that if a position only matches the discovered motif poorly, then $poff_{ij}$ 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 $(poff_{ij})$ 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 $poff_{ij}$ value above that threshold were completely erased.

3 Experimental results

This section describes experiments using MEME that were conducted on two datasets. The first, which will be referred to as the CRP/LexA dataset, comprises DNA fragments which contain examples of 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, III, 1989] who, in turn, derived it from [Berg and von Hippel, 1988] and [de Crombrugghe *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 homology 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 homology, 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.

dataset	samples	length of samples	proven CRP sites	proven LexA sites
CRP	18	105	18	0
LexA	16	(approx.) 200	1	11
CRP/LexA	34	(average) 150	19	11
promoter	231	60	NA	NA

Table 1: Overview of the contents of the datasets.

3.1 MEME can discover two different binding site motifs

MEME was run for 5 passes on the CRP/LexA dataset with site $LSITE = 20$, $MAXP = 17$, $NITER = 1$. The value for $LSITE$ 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.[4] The value for $MAXP$ 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 $MAXP$ 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 $poff_{ij}$ 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 $poff_{ij}$ for each sample in the dataset after each pass. Table 3 shows the values of $poff_{ij}$ after pass 1 of MEME for the known LexA binding sites. It can be easily seen that the model found in the first pass characterizes the LexA site. Furthermore, all other values of $poff_{ij}$ were below 0.17, so the model appears to be very specific for the LexA site.

---

[4]If the best value of $LSITE$ 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 $LSITE$. Each run of MEME uses a single value of $LSITE$ for all motifs found.

pass	starting subsequence	final consensus	relative entropy	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

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 *freq* matrix by choosing the letter with the highest probability. The values of relative entropy 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 relative entropy than the models found on later passes. Note that $LSITE = 20$ and $MAXP = 17$.

The consensus pattern for the model discovered in pass 1 of MEME on the CRP/LexA dataset also agrees exceedingly well with the LexA site. MEME prints the consensus (i.e. the most probable letter for each position in the motif as determined from *freq*) 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 relative entropy score of each subsequence of the input data. This is done using the specificity matrix method of [Hertz *et al.*, 1990]. Figure 3 shows the relative entropy 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 pattern it reports for pass 2 is TTTTTTGATCGGTTTCACAC, which agrees well with the consensus found by the original EM algorithm and reported in [Lawrence and Reilly, 1990]. More significantly, the model characterizes the CRP motif well, judging from the values of $poff_{ij}$ for the various positions in the samples in the dataset. Table 4 shows the values of $poff_{ij}$ found during pass 2 on the CRP/LexA dataset. According to [Lawrence and Reilly, 1990],

sample	site 1	$poff_{ij}$	site 2	$poff_{ij}$	site 3	$poff_{ij}$
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 homology.

Table 3: Values of $poff_{ij}$ 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 $poff_{ij}$ 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 $poff_{ij}$ of the model at those positions. All other positions have values of $poff_{ij}$ below 0.17. Although the site at position 112 in the colicin E1 sequence has $poff_{ij}$ value only 0.05, this is one of the four highest $poff_{ij}$ values for this sequence. No proven sites are known for *himA* and *uvrC* and $poff_{ij}$ for all positions in those samples was very low, less than 0.0001.
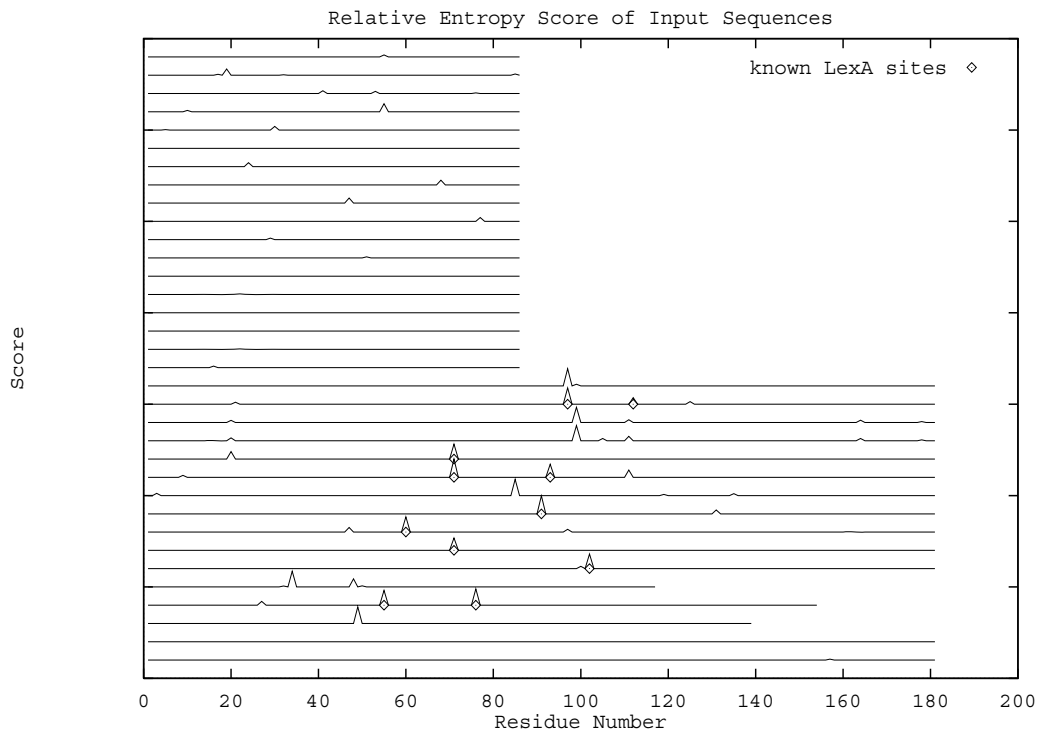
Figure 3: The relative entropy score of each subsequence of the CRP/LexA dataset using the specificity matrix of pass 1 of MEME. The CRP samples are the short curves at the top, while the LexA samples are the long curves at the bottom. Vertical scale is such that highest peak is 24.3 bits. All values below zero have been set to zero.

15

the CRP dataset contains 24 known CRP binding sites, 18 of which had been verified by protection experiments. The value of $poff_{ij}$ for eight of these is above 0.99 in the model, while eleven have $poff_{ij}$ 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 $poff_{ij}$ is over 0.999, which is extremely high. Because of the overrepresentation 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 4 shows the relative entropy 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 site $LSITE = 6$, $MAXP = 231$, $NITER = 1$. The value for $LSITE$ 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 $MAXP$ 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 known -35 region consensus, `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 relative entropy plotted in Figures 5 and 6. The residue 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 5 shows that the model identifies the -10 consensus region of the promoters. A column of peaks at position 15 of Figure 6 confirms that the second model identifies the -35 region of the promoters, even though its consensus sequence is slightly different

16

sample	site 1	$poff_{ij}$	site 2	$poff_{ij}$
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 homology.

[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
in [Varley and Boulnois, 1984] and [Hertz et al., 1990].

Table 4: Values of $poff_{ij}$ 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 $poff_{ij}$, and twelve more (those not stated as < some bound) have
values of $poff_{ij}$ among the top four $poff_{ij}$ 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 may have been previously reported as being a LexA
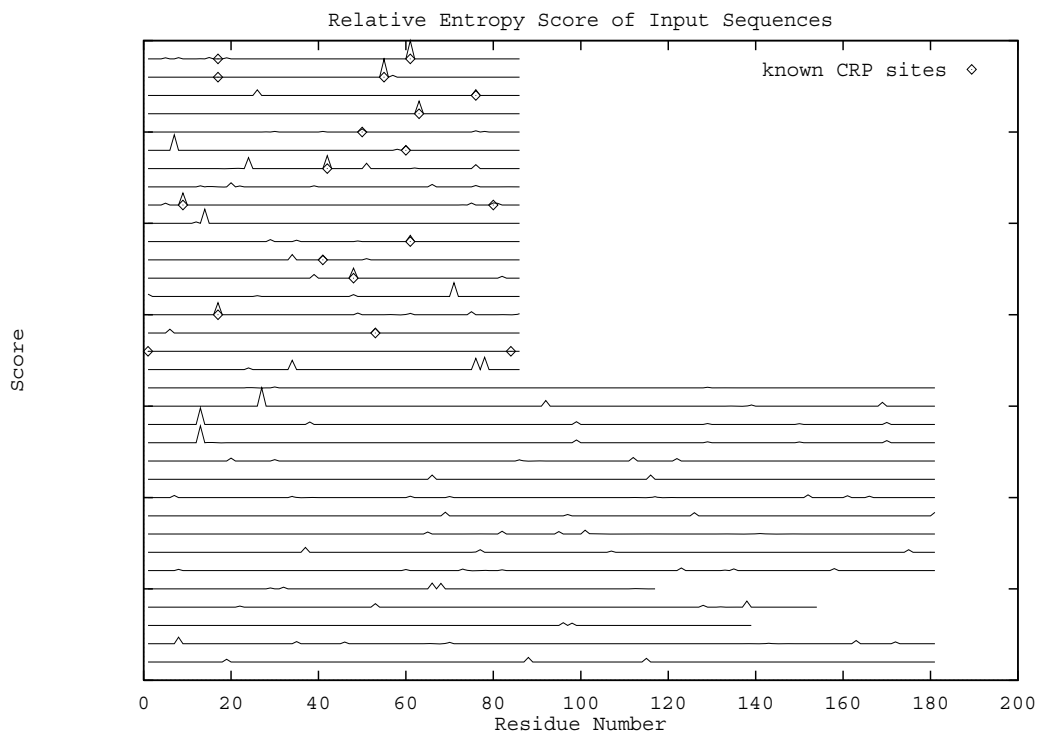site.

Figure 4: The relative entropy score of each subsequence of the CRP/LexA dataset using the specificity matrix of 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.

pass	starting subsequence	final consensus	relative entropy	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

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 and have much higher log(_likelihood_) and relative entropy than the other models found.

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.

4.1 Subsequence-derived starting points work well with EM

The idea of running EM for only a few iterations 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 the MEME version of EM for one or two iterations 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 a couple of iterations. (Then EM could be run to
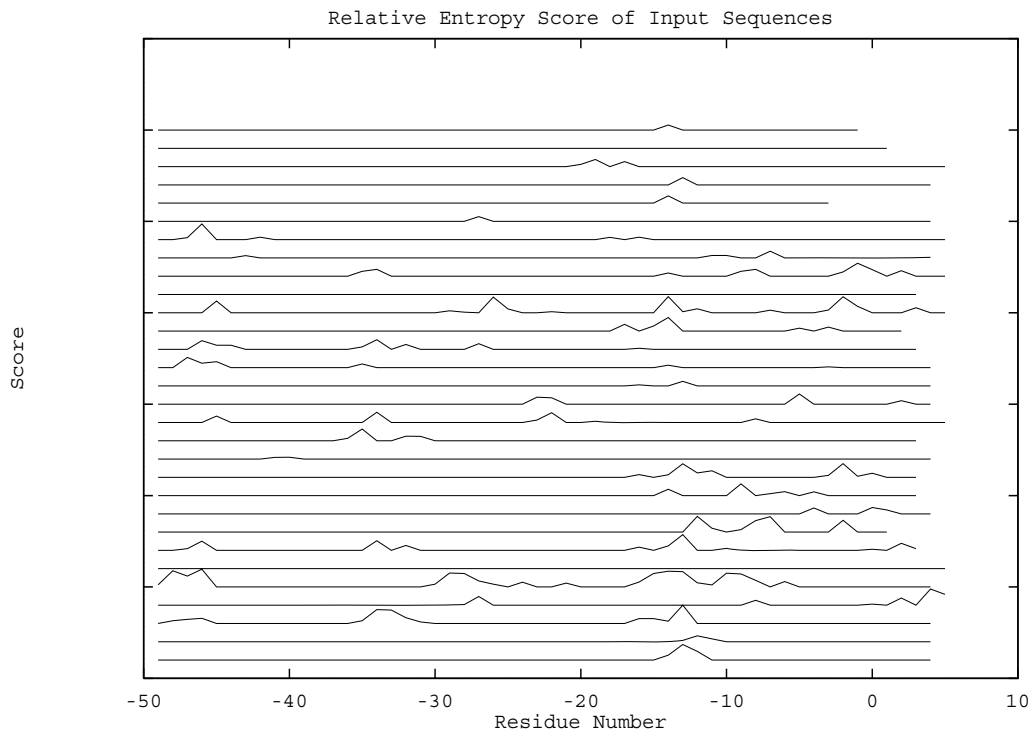
Figure 5: The relative entropy score of each subsequence of the promoter dataset using the specificity matrix of pass 1 of MEME. The concept learned on pass 1 of MEME on the promoter dataset locates the −10 region of the promoters. The vertical scale is such that highest peak is 7.21 bits. All values below zero have been set to zero.
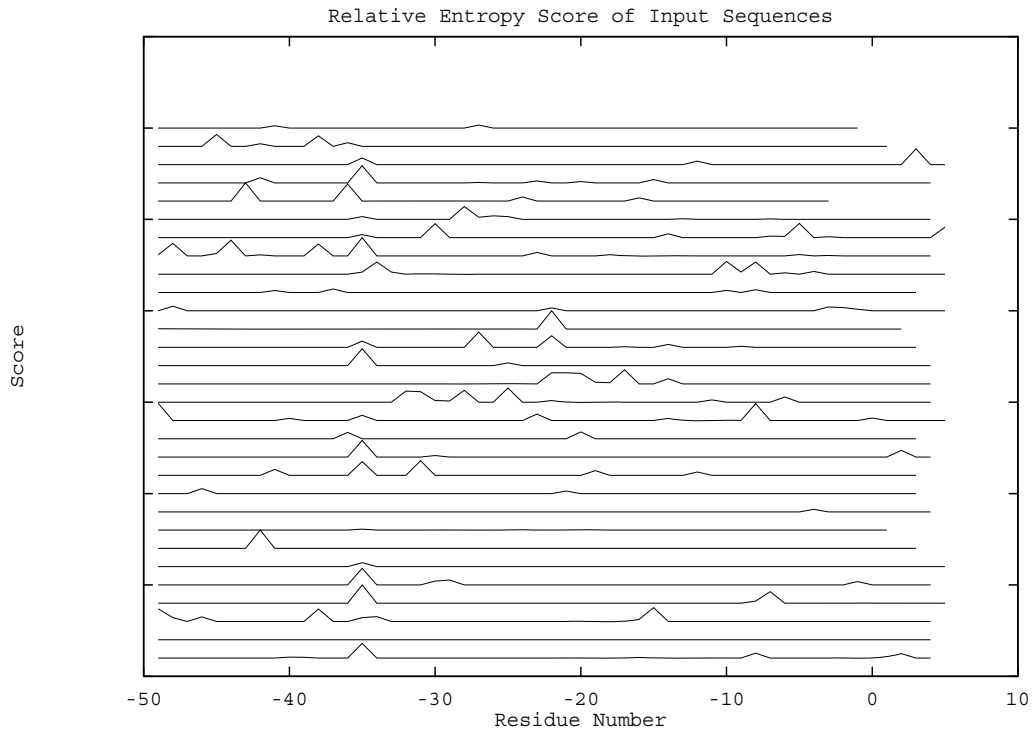
Relative Entropy Score of Input Sequences

Figure 6: The relative entropy score of each subsequence 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.
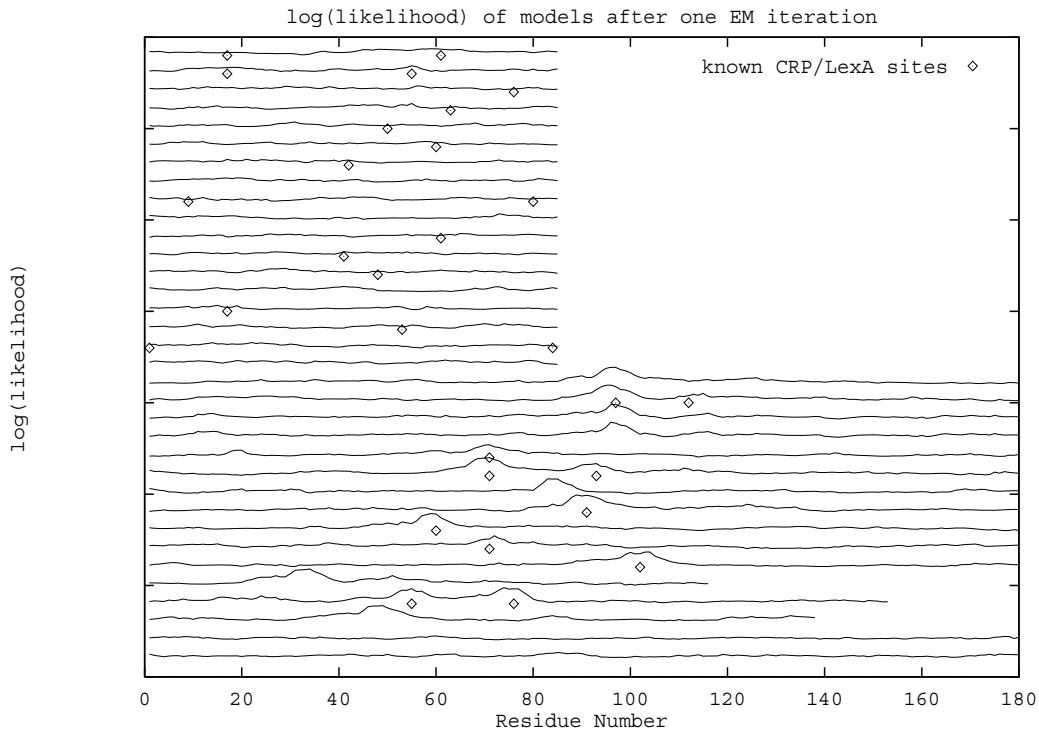
21

Figure 7: log(*likelihood*) after one iteration of modified 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 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.

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 that large peaks in the likelihood function were occurring in most of the LexA samples (Figure 7).

Further investigation showed that the peaks tended to occur at the positions of the known LexA sites. Figure 8 shows an expanded view of the curve for the sample from recN. The recN sample contains three LexA sites whose 5' ends are marked on the
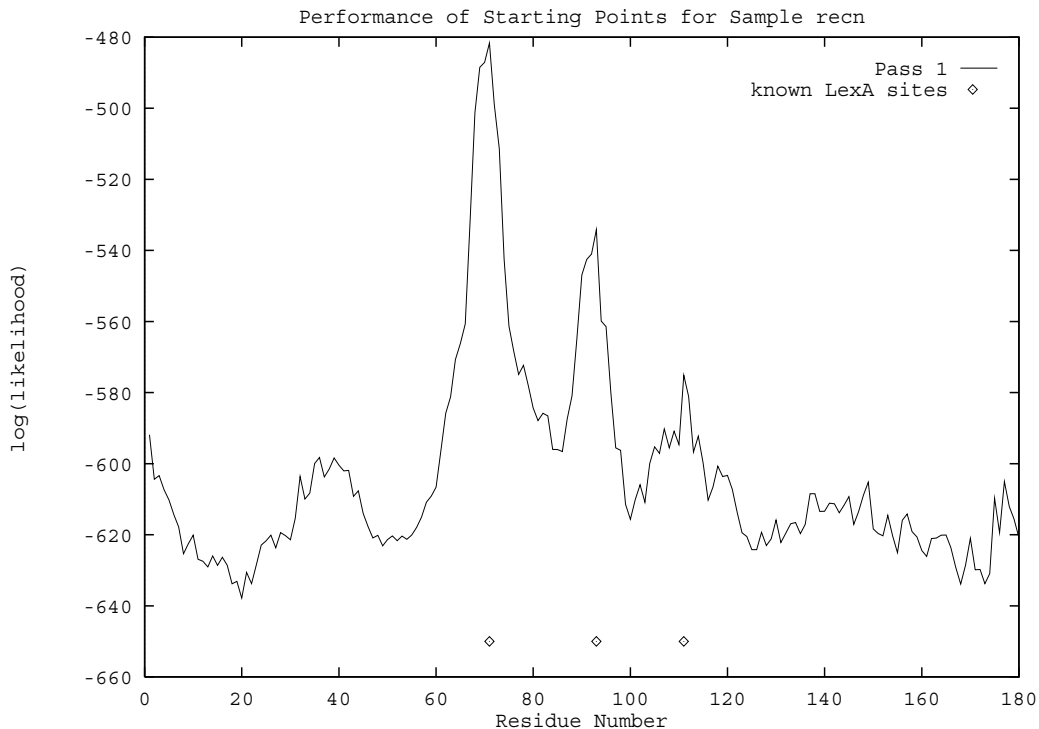
Figure 8: Modified EM finds models of high likelihood when run for one iteration on the CRP/LexA dataset from starting points derived from subsequences of sample recN. The starting points correspond well with the known LexA sites, whose 5' ends are indicated on the horizontal axis.

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 sites. Figure 9 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 sites from other peaks which do not correspond to any known sites. It appears that the other peaks are due to modified EM starting to converge to a model related to the LexA data. Even
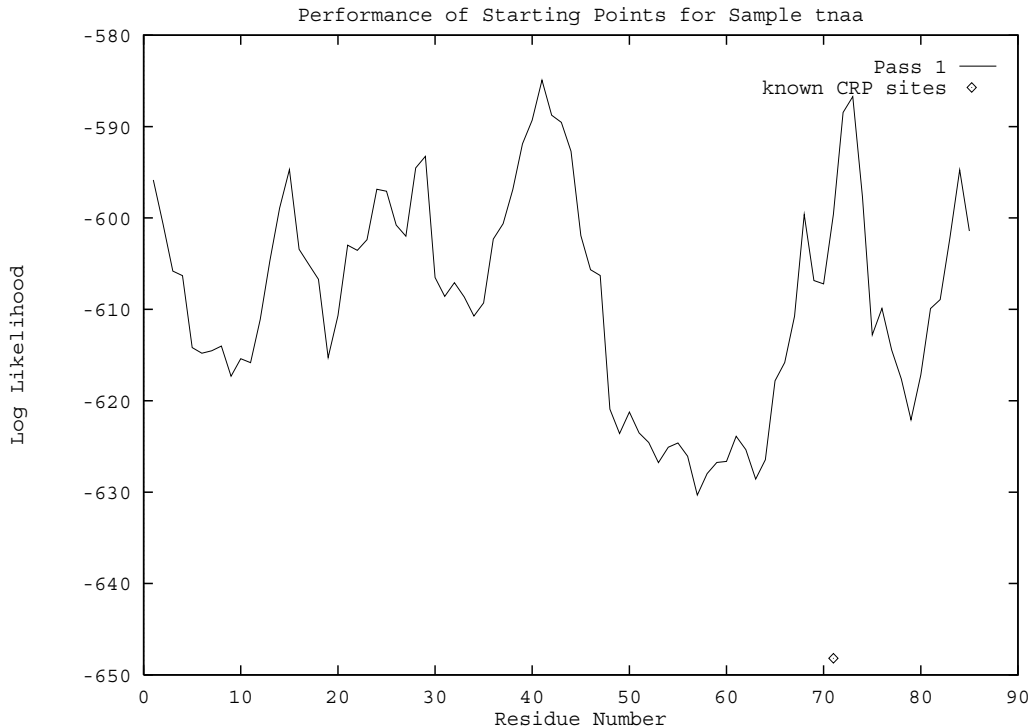
Figure 9: The log(*likelihood*) of the model after 1 iteration of EM in MEME varies strongly with the starting point.

a bad model of the highly conserved LexA motifs may have log(*likelihood*) similar to the best model of the CRP sites, due to the fact that the LexA sites are much more highly conserved than the CRP 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 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 sites from the data in order to be able to discover the best starting points from which to run modified EM to learn a model for the CRP sites.

4.3 The expected number of motif appearances is not critical

If the choice of *MAXP* were critical to the ability of MEME 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 *MAXP* set to a wide range of values. So running

MEME with just a few values of $MAXP$ will probably suffice to find most motifs (if any) which are represented in a dataset. Also, the relative entropy of the models is typically maximum when the value of $MAXP$ is near the "correct" value. This phenomenon can provide a quick estimate of how many appearances of a motif are present in the dataset.

MEME was run on the CRP/LexA dataset with various values of $MAXP$ and all other parameters fixed. The models found by MEME on each pass were examined by eye to see if they 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 relative support and $\log(likelihood)$ of the models. MEME always finds a model for the LexA motif on the first pass. With low $MAXP$, it finds LexA more than once, due presumably to the fact the LexA sites do not get completely erased. (MEME effectively erases at most $MAXP$ occurrences of a site after each pass, so if $MAXP = 5$ and there are fifteen LexA 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 $MAXP$ tried except for $MAXP = 5$. Usually, CRP was the second model found. While the values of relative entropy 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 $MAXP$ was close to the actual number of known CRP sites in the dataset was the relative entropy and $\log(likelihood)$ of the CRP model much higher than for the other models (of unknown significance) found by MEME.

The relative entropy of the best models found by MEME for the LexA and CRP binding sites varied as a function of the value of $MAXP$ used. It can be seen that the relative entropy for the models found of both binding sites tends to peak when $MAXP$ is close the actual number of sites in the CRP/LexA dataset. This peaking behavior may be a clue for distinguishing models which describe real sites from background noise. The peak in the curve labeled "other" in Figure 10 corresponds to the consensus GTGAAAGACTGTTTTTTTGA which was found both with $MAXP = 5$ and $MAXP = 10$. It is interesting to speculate that this might be an actual binding site. However, higher values of $MAXP$ did not yield models similar to it on any of the first five passes of MEME.

MAXP	pass	consensus	relative entropy	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	?

Table 6: MEME finds models of the LexA and CRP binding sites when *MAXP* has values between 10 and 35. When *MAXP* is above 10, LexA and CRP are usually found on the first two passes. Only with *MAXP* = 5 did MEME fail to find CRP on any of the first five passes.
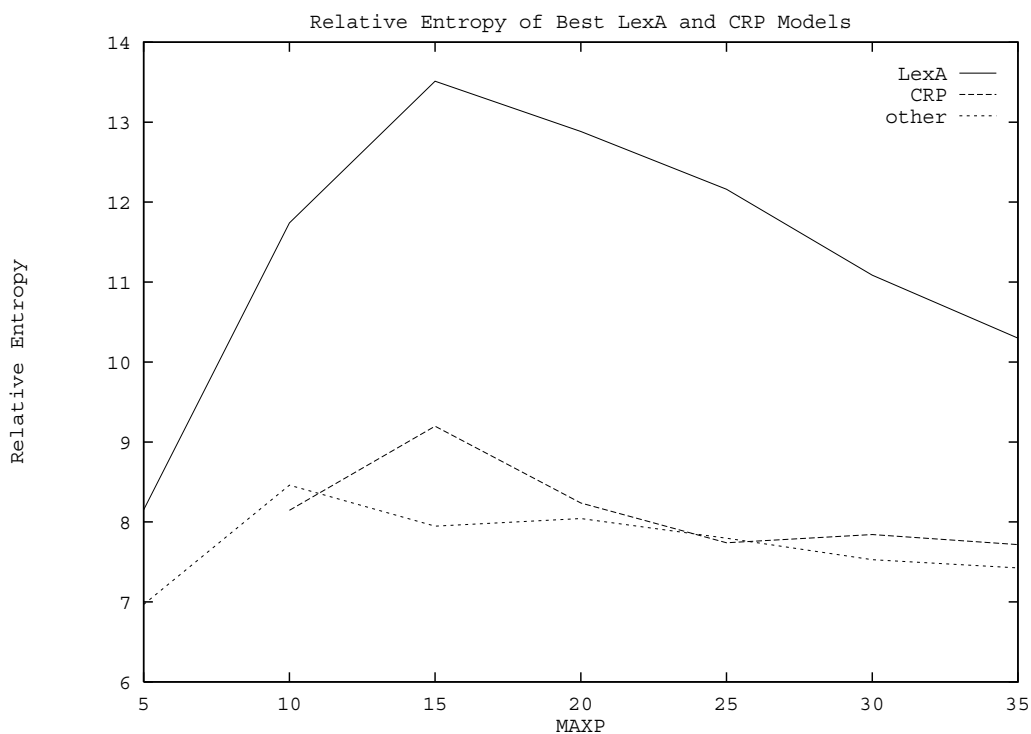
Figure 10: The relative entropy of the best LexA and CRP models found by MEME for different values of $MAXP$ peaks around $MAXP = 15$. The highest relative entropy of a model which did not characterize either LexA or CRP binding sites is plotted to show that although LexA always stands out, CRP only does so when $MAXP$ is close to the actual number of CRP sites in the dataset. MEME was run on the CRP/LexA dataset with $LSITE = 20$, $NITER = 1$, $NPASSES = 5$.

4.4 MEME is less sensitive to noise than standard EM

The removal of the one-motif-appearance-per-sequence assumption was intended, among other things, to make MEME less sensitive to noise than EM. To test this assumption, both MEME and standard EM were run on datasets which contained varying numbers of randomly generated sequences. The random sequences had the same base 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 learned the correct concept, on the first pass, from datasets with more random sequences than the standard EM algorithm could tolerate. MEME 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.) EM was not able to learn a LexA binding site model with more than 60 random samples added to the dataset. (It learned it "off-center" when more than 20 random samples were in the dataset.) MEME, however, learned the correct LexA model even with 80 random samples added to the dataset.

Figure 11 shows the relative entropy of the CRP and LexA models learned by MEME and EM on the first pass from datasets with various numbers of random sequences added. The CRP models learned by MEME also consistently had higher relative entropy than those learned by standard EM. This was true even for the model learned with no random sequences added to the dataset. Presumably, this is indicative of the fact that MEME is taking advantage of the sequences with multiple appearances of the CRP site. The models learned by MEME for LexA were extremely robust to the number of random samples added to the dataset. There was almost no decrease in the relative entropy no matter how many random samples were present. EM, on the other hand, found models with lower relative entropy the more random samples were in the dataset.

It is clear from Figure 11 that MEME will find a set of highly conserved binding sites even in datasets where the vast majority of the sequences do not contain it. The original version of EM suffers from the fact that it must always average in one supposed motif appearance from each sample. Even if the original version of EM were run for several passes, the model of the motif found would have low relative entropy and be hard to distinguish from other models not representing actual binding sites.
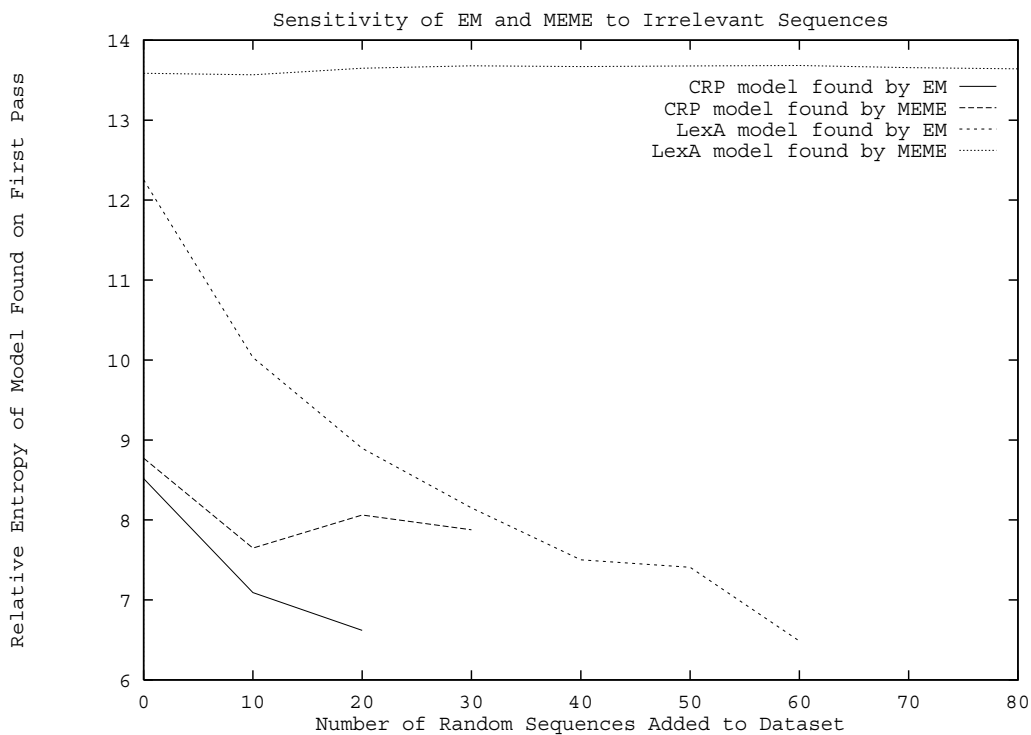
Figure 11: The relative entropy of the LexA and CRP models found by EM and by the first pass of MEME, run separately on the CRP and LexA datasets with different numbers of random examples added. The comparative advantage of MEME is clear. Especially with motifs whose occurences are highly conserved, MEME finds very good models even when many sequences not containing the motif are present. MEME was run with $LSITE = 20$, $MAXP = 15$, $NITER = 1$, $NPASSES = 1$.

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 basic EM method which allow MEME to drop the assumption that each sequence contains exactly one appearance of a motif have been shown to give MEME the ability to discover motifs in datasets which contain larger numbers of 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, relative entropy 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 relative entropy scores of various positions of the sequences in the dataset such as in Figure 5 and Figure 6 can also be helpful to the 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 MEME, eliminates the assumption of one motif per sequence. It may also be possible to adapts MEMEś 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 by [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 relative support 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.

References

[Bailey, 1993] Timothy L. Bailey. Likelihood vs. information in aligning biopolymer sequences. Technical Report CS93-318, University of California, San Diego, February 1993.

[Berg and von Hippel, 1988] Otto G. Berg and Peter H. von Hippel. Selection of dna binding sites by regulatory proteins. *Journal of Molecular Biology*, 200:709–723, 1988.

[Breiman *et al.*, 1984] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

[Cardon and Stormo, 1992] Lon R. Cardon and Gary D. Stormo. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *Journal of Molecular Biology*, 223:159–170, 1992.

[de Crombrugghe *et al.*, 1984] Benoit de Crombrugghe, Stephen Busby, and Henri Buc. Cyclic AMP receptor protein: Role in transcription activation. *Science*, 224:831–838, May 1984.

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

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

[Haussler *et al.*, 1993] David Haussler, Anders Krogh, I. Saira Mian, and Kimmen Sjolander. Protein modeling using hidden Markov models: Analysis of globins. In *Hawaii International Conference on System Sciences*, 1993.

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

[Lawrence and Reilly, 1990] Charles E. Lawrence and Andrew A. Reilly. 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, 1990.

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

[Sakakibara *et al.*, 1993] Yasubumi Sakakibara, Michael Brown, Rebecca C. Underwood, I. Saira Mian, and David Haussler. Stochastic context-free grammars for modeling RNA. Technical Report 93-16, UCSC-CRL, June 1993.

[Stormo and Hartzell, III, 1989] Gary D. Stormo and George W. Hartzell, III. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 86:1183–1187, 1989.

[Varley and Boulnois, 1984] Jennifer M. Varley and Graham J. Boulnois. Analysis of a cloned colicin Ib gene: complete nucleotide sequence and implications for regulation of expression. *Nucleic Acids Research*, 12(17):6727–6739, 1984.

A Calculating *freq* and *poff* in EM and MEME

During each iteration of EM, the values of the letter probabilities of the motif model, *freq*, and of the offset probabilities, *poff*, must be reestimated. Given *freq*, the *poff* values are reestimated using Bayes' rule. Given the values of *poff*, *freq* is estimated as the expected values of the letter frequencies. How this is done is described here for both standard EM as well as for the modified EM algorithm used by MEME.

To describe the EM and MEME algorithms formally, the following definitions are useful. Let N be the number of sequences, $LSITE$ be the length of the motif, and L be the length of each sequence (assume all are of the same length). Define $poff_{ij}^{(q)}$ 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 $freq_{lk}^{(q)}$ 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.

In the probabilistic model we use, we ignore the probability of the letters outside of the motif, and only consider the probability of the letters in the motif. Both standard and modified EM must calculate the probability of sequence S_i given the motif start and the model. This can be written as

$$P(S_i|Y_{ij} = 1, poff^{(q)}) = \prod_{k=1}^{LSITE} freq_{l_k,k}$$

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 $poff^{(q)}$.

The standard EM algorithm uses Bayes' rule to estimate $poff^{(q)}$ from $P(S_i|Y_{ij} = 1, poff^{(q)})$. Bayes' rule states that

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

so

$$poff_{ij}^{(q)} = P(Y_{ij} = 1 \mid freq^{(q)}, S_i) = \frac{P(S_i|Y_{ij} = 1, freq^{(q)})P^0(Y_{ij} = 1)}{\sum_{k=1}^{L-LSITE+1} P(S_i|Y_{i,k} = 1, freq^{(q)})P^0(Y_{i,k} = 1)}$$

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,

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

so the above simplifies to

$$poff_{ij}^{(q)} = \frac{P(S_i | Y_{ij} = 1, freq^{(q)})}{\sum_{k=1}^{L-LSITE+1} P(S_i | Y_{i,k} = 1, freq^{(q)})}$$

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

Notice that the above formula for $poff^{(q)}$ ensures that it sums to 1.0 for each sequence. This enforces the implicit assumption of the original EM algorithm that each sequence contains exactly one appearance of the shared motif. Our modified version of EM normalizes $poff^{(q)}$ so that the sum over all positions in all sequences is $MAXP$. This can be written formally as

$$poff_{ij}^{(q)} = MAXP \frac{P(S_i | Y_{ij} = 1, freq^{(q)})}{\sum_{n=1}^{N} \sum_{k=1}^{L-LSITE+1} P(S_n | Y_{i,k} = 1, freq^{(q)})}$$

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

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

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

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 thus is the only or best choice. The two constraints are enforced separately by applying the following two algorithms in order. Figure 12 presents the first algorithm, which makes one or more passes through the offset probabilities normalizing them to sum to $MAXP$ 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

34

the value of offset probabilities that have never been squashed so that the $MAXP$ total is enforced. In practice, usually few passes are needed. The second algorithm, given in Figure 13 is run next to enforce the constraint that no window of $LSITE$ positions has offset probabilities that sum to more than 1.0. This is achieved by dividing each sequence into adjacent windows of length $LSITE$ and normalizing within each window separately. Windows are then shifted one to the right and the process is repeated. This is done for all $LSITE$ possible shifts of the windows, which guarantees that no window of width $LSITE$ will have offset probabilities summing to greater than 1.0, but may reduce the total sum below $MAXP$. The squashing algorithm could be repeated to correct this but this is not done in the interest of saving computation time.

INPUT:

    poff (unnormalized; $poff_{ij}^{(q)} = P(S_i|Y_{ij} = 1, freq^{(q)})$)

    total (the total of $poff^{(q)}$ for all sequences and positions)

    $MAXP$ (the number of appearances of the motif expected)

    L (length of the sequences)

    N (number of sequences)

ALGORITHM:

    renormalize = true

    while renormalize

        renormalize = false

        normalize = total / $MAXP$

        total = 0

        for i=1 to N

            for j=1 to $L - LSITE + 1$

                p = $poff_{ij}$

                if p < 1

                    p = p/normalize

                    if p > 1

                        p = 1

                        $MAXP = MAXP - 1$

                        renormalize = true

                    end

                end

                $poff_{ij}$ = p

                if (p < 1) total = total + p

            end

        end

Figure 12: SQUASH: Constrain all offset probabilities to be ≤ 1.0.

INPUT:
   $poff^{(q)}$ (normalized offset probabilities)
   L (length of the sequences)
   N (number of sequences)

ALGORITHM:
   for i=1 to N
      for offset=1 to $LSITE$
         for j=offset to $L - 2 * LSITE$ by $LSITE$
            localp = 0
            for k=1 to $LSITE$
               localp = localp + $poff^{(q)}_{i,j+k}$
            end
            if localp > 1
               for k=1 to $LSITE$
                  $poff^{(q)}_{i,j+k}$ = $poff^{(q)}_{i,j+k}$ / localp
               end
            end
         end
      end
   end

Figure 13: SMOOTH: Constrain the sum of offset probabilities in any window of width $LSITE$ to sum to no more than 1.0.