Detecting Motifs from Sequences

Yuh-Jyh Hu[1], Suzanne Sandmeyer[2], and Dennis Kibler[1]
Information and Computer Science Department[1]
Biological Chemistry Department, College of Medicine[2]
University of California, Irvine
yhu@ics.uci.edu, sbsandme@uci.edu, kibler@ics.uci.edu

Abstract

The problem of multiple global comparisons of families of biological sequences has been well-studied. Fewer algorithms have been developed for identifying local consensus patterns or motifs in biological sequences. These two important problems have different biological constraints and, consequently, different computational approaches. The difficulty of finding the biologically meaningful motifs results from the variability in (1) the bases at each position in the motif, (2) the location of the motif in the sequence and (3) the multiplicity of motif occurrences within a given sequence. In addition the short length of many biologically significant motifs and the fact that motifs gain biological significance only in combinations, makes them difficult to determine using standard statistical methods. In this paper we introduce our own approach, DMS, which combines multiple objective functions with an improved iterative sampling search method. We compare the main approaches for finding motifs and test the effectiveness of the various algorithms by comparing them on ten real domains and fourteen artificial domains. The main advantage of DMS is that it is better able to find shorter motifs.

1 Introduction

Genome projects are generating large data sets of genomic sequence data. However, the size and speed of acquisition of these data sets exceeds the pace of experimental analyses and interpretations. In 1996, with the international collaboration of over 100 laboraties, the yeast genome was completely sequenced. It has 12 million base pairs (bps) and about 6,000 genes. To the surprise of biologists, the biological functions of only about 2,000 genes were known. The functions of another 2,000 genes could be inferred by comparison. The functions of the remaining 2,000 genes, called orphans, are unknown. Recently the complete genome (approximately 100 million bps) of a multi-celled animal (*C. elegans*) was sequenced. Within a few years the sequencing of the human genome (approximately 3 billion bps) is anticipated. Once the genome and genes have been determined, there are two essential questions to be answered: 1) What is the function of each gene, and 2) When are genes expressed?

The first question has been heavily studied and primarily depends on both experimental tests of gene activities and predictions of gene activity base on structure. Computationally, the most successful way of characterizing a gene has been based on probabilistic models, usually via some instantiation of Hidden Markov Models (HMMs). HMMs work well for this problem since they provide a global model which allows insertions, deletions, and substitutions. These capabilities match the intuition that similar genes have had a common evolutionary history and the evolution process involves insertions, deletions and changes to the base pairs.

The second question has been less well studied and has a very different character. Biologists have determined that the control or regulation of gene expression is primarily determined by relatively short sequences in the region surrounding a gene. These sequences vary in length, position, redundancy, orientation, and bases. Insertions and deletions are uncommon in regulatory motifs. These qualities prohibit the simple application of HMMs.

Several methods have been developed for detecting patterns shared by functionally related biosequences (Helden *et. al.*, 1998; Hertz & Stormo, 1995; Hertz *et. al.*, 1990; Bailey & Elkan, 1995; Lawrence *et. al.*, 1993; Hughey and Krogh, 1996; Eddy, 1995). We review these methods according to their representations, objective functions, and search strategies.

In addition, we present a new approach called DMS to detect motifs from sequences. DMS extends previous work by combining multiple objective functions with an improved iterative sampling technique. The performance of our algorithm and several others are compared on ten problems taken from the biological literature. Each of these problems consists of a set of biological sequences with known motifs. To further understand the limitations and value of these programs, we also compared the programs on fourteen artificial problems, which were designed to mimic real data.

2 Characteristics of Motif-finding Problem

Fundamentally gene regulation is determined by chemical reactions which are, in turn, controlled by the shape and electrostatic charges of the molecules involved. One such instance of this is the interaction between regulatory proteins and their target binding site. The significance of this is that this can lead to a coordination of regulation via a combination of motifs. Unfortunately this information is not typically available. We expect that the local shape of a binding or receptor site will be primarily determined by the bases involved, acknowledging the fact that non-local base changes can affect local shape.

There are a number of consequences that we can expect from this view. These consequences are supported by the structure of known motifs.

- Patterns are relatively short since they only define a local shape.
- Patterns are not defined by an exact sequence of bases, and variation is allowed. Typically the variation is represented via a probability matrix.
- The precise location of the receptor site may not be important, as the goal of the receptor site is to bind to another molecule.
- Multiple occurrences of a receptor site may be important since each occurrence would give a molecule a greater chance of finding the binding site and since bound molecules may interact, modifying the structure and the binding properties.
- Insertion and deletions are less likely to occur than base variation, as this would have a drastic effect on the conserved features of the receptor.
- The pattern or motif should be common to most of these sequences in a family and uncommon in the entire genome. It is essential that not all genes are expressed, but only a selected few. Also there may be multiple ways of turning on a gene, so it is not required that the motif occur in every sequence in a given family.

These characteristics make the problem somewhat ill-defined. The terms of "common", "pattern", and "most" require precise definitions. While various definitions are possible, which best corresponds to the underlying biological problem is unclear.

In any case these characteristics make the problem computationally difficult. For example, a typical problem would be: given 30 DNA sequences, each of length 800, find a common pattern of length 8. Let us simplify the problem, as many algorithms do, and assume the pattern occurs exactly once in each sequence. This means that there are approximately 800^{30} potential locations for a motif candidate.

3 Issues in Motif-Finding Algorithms

There are three main interrelated computational issues: the representation of a pattern, the definition of the objective function, and the search strategy. While we examine the algorithms on computational grounds, the final, gold-standard is how well the algorithm does at predicting motifs.

3.1 Representation

As the primary DNA sequences are described by a double-stranded string of nucleic bases {A,C,G,T}, the most basic pattern representation is the exact base string. Due to the complexity and flexibility of the motif binding mechanism, there is rarely any motif that can be exactly described by a string of nucleic bases. To obtain more flexibility, the IUPAC code was designed, which extends the expressiveness of the simple base string representation by including all disjunctions of nucleotides. In this language there is a new symbol for each possible disjunction, e.g. W represents A or T.

A more informative pattern representation is a probability matrix in which each element reflects the importance of the base at a particular position. Such matrices can be easily translated into the IUPAC code, while the converse is not true. These matrices are often transformed from the observed occurrence frequencies. One limitation of probability matrices is that correlation or dependence between positions are not represented.

3.2 Objective Function

The purpose of an objective function is to approximate the biological meanings of the patterns in terms of a mathematical function. The objective function are heuristics. Once the objective function is determined, the goal is to find those patterns with high objective function value. Different objective func-

tions have been derived from the background knowledge, such as the secondary structures of homologous proteins, the relation between the energetic interactions among residues and the residue frequencies, etc (Stormo, 1990; Lawrence *et. al.*, 1993). Objective functions based on the information content or its variants were proposed (Hertz *et. al.*, 1990; Lawrence *et. al.*, 1993). Others evaluate the quality of the pattern by its likelihood or by some other measures of statistical significance (Bailey & Elkan, 1995; Helden *et. al.*, 1998). In addition, some define the pattern as a model of a probabilistic sequence generator and evaluate the model by the probability that the given sequence data is generated by the model (Hughey and Krogh, 1996; Eddy, 1995).

Even though there are many different objective functions currently used, it is still unclear what is the most appropriate object function or the best representation for patterns that will correspond to biological significant motifs. More likely, additional knowledge will need to be incorporated to improve motif characterization. In the final analysis, the various algorithms can only produce candidate motifs that will require biological experiments to verify.

3.3 Search Strategy

If one adopts the exact string representation, then one can exhaustively check every possible candidate. However this approach is only able to identify short known motifs or partial long motifs (Helden *et. al.*, 1998). Therefore, the primary representation used is a probability matrix (Harr *et. al.*, 1983; Staden, 1984; Hertz *et. al.*, 1990; Lawrence *et. al.*, 1993; Bailey and Elkan, 1995). Once one accepts a probability matrix as the representation, then there is no possibility for an exhaustive search. Initial approaches started with hill-climbing strategies, but these typically fell into local optimum. Standard approaches to repairing hill-climbing, such as beam and stochastic search, were tried next. The current approaches involve a mixture of sampling and stochastic iterative improvement. This avoids the computational explosion and maintains or improves the ability to find motifs (Lawrence *et. al.*, 1993; Bailey and Elkan, 1995).

4 The DMS Algorithm

DMS adopts the probability matrix representation for motifs. The user provides a family of sequences and how many motifs he would like returned. The system returns that number of motifs, ranked by a significance measure, which will be defined.

The probability matrix representation has been used in various pattern identification problems (Harr *et. al.*, 1983; Staden, 1984; Hertz *et. al.*, 1990; Lawrence *et. al.*, 1993). It is usually built from the base frequency of example biosequences. For example, in the NIT regulatory family (Helden, *et al.*,1998) which contains 7 members, a possible 6-base motif matrix is illustrated in Figure 1. The normalized matrix is also shown in this figure.

Based on the normalized motif matrix, we could calculate the match score of any 6-base sequence by dividing the sum of the values for each position of the motif. For example, given a 6-base sequence, GATAAG, its match score is $\frac{0.86+1+1+1+1+1}{6}$. The success of these analyses confirms the fact that the frequencies of bases at positions within sites are related to the importance of the bases to the functioning within the sites (Stormo, 1988). The challenge is to find a matrix that well represents the motif in terms of the objective function.

We propose a new motif-finding algorithm, DMS. Unlike other approaches, DMS uses multiple types of objective functions, the motif consensus quality, the motif multiplicity significance and the motif coverage. The consensus quality is only used to guide the search for well-conserved motif candidates, the motif multiplicity significance reflects the value of multiple copies of motifs, and the motif coverage addresses the importance of a motif's being commonly shared by a given family of sequences.

The consensus quality of a matrix is derived from the entropy, the lower the entropy, the better conserved the motif. The entropy is calculated from the probability that each base occurs at each position in the motif, Pm_{base}. More precisely, the entropy for a particular column n in the matrix is given by:

$$E_n = -\sum_{i=b_1}^{b_4} Pm_i \lg Pm_i$$

where $b_1..b_4$ are the bases A, G, C, T. If the bases are uniformly distributed over a position, then the maximum value of 2 is obtained. If only a single base appears in a position then the minimum value of 0 is obtained. Thus we define the consensus quality of column n as:

$$C_n = 2 - E_n$$

The final consensus quality of a matrix b, is defined as the average of all position quality.

$$con(b) = \frac{1}{W} \sum_{n=1}^{W} C_n$$

where W is the width of the motif.

The multiplicity significance is derived from the measure of precision as defined in the information retrieval

```
A 0 7 0 7 7 0                    A 0.00 1.00 0.00 1.00 1.00 0.00
G 6 0 0 0 0 7  normalized to      G 0.86 0.00 0.00 0.00 0.00 1.00
C 1 0 0 0 0 0                    C 0.14 0.00 0.00 0.00 0.00 0.00
T 0 0 7 0 0 0                    T 0.00 0.00 1.00 0.00 0.00 0.00
```

Figure 1: A 6-base Motif Matrix Example

paradigm. It is simple and empirically effective. We define the multiplicity significance of a motif b as:

$$mul(b) = \frac{occ_S(b)}{occ_G(b)}$$

where $occ_S(b)$ is b's occurrences in S, $occ_G(b)$ is b's occurrences in genome.

The motif coverage is defined as the ratio of the number of the sequences containing b to the total number of sequences given, i.e.:

$$cov(b) = \frac{cont_S(b)}{|S|}$$

where $cont_S(b)$ is the number of sequences in S that contain b, and $|S|$ is the total number of sequences in S.

Given a set of N biosequences, DMS carries out an iterative improvement search strategy that attempts to find a user-defined number, d, of matrices that maximize the consensus quality defined above. These d matrices are motif candidates. DMS then ranks these motifs according to a merit measure based on the combination of the consensus quality, the multiplicity significance and the motif coverage. Given the d motifs, we first normalize the consensus quality, the multiplicity significance and the motif coverage of each motif b, using the maximum value, as defined below :

$$Con_{normal}(b) = \frac{con(b)}{MAX(con)}$$
$$Mul_{normal}(b) = \frac{mul(b)}{MAX(mul)}$$
$$Cov_{normal}(b) = \frac{cov(b)}{MAX(cov)}$$

where $MAX(con)$ is the maximum consensus quality of the d motifs, $MAX(mul)$, the maximum multiplicity significance of the d motifs, and $MAX(cov)$, the maximum motif coverage of the d motifs.

We assign the equal weight to every measure discussed above and propose a final merit measure of a motif b defined as :

$$Merit(b) = \frac{1}{\frac{1}{3}\left(\frac{1}{Con_{normal}(b)} + \frac{1}{Mul_{normal}(b)} + \frac{1}{Cov_{normal}(b)}\right)}$$

The value of merit reflects the synergy of the consensus quality, the multiplicity significance and the motif coverage. There are three steps in DMS which are detailed in the following subsections.

4.1 Translation: Subsequences into Matrices

If we knew the motif location(s) in every sequence, we could start with a probability matrix corresponding to these positions. As this is unknown, we begin by allowing each subsequence of length W to be a candidate motif. We convert this particular subsequence into a probability matrix in two steps, adopting an idea from (Bailey and Elkan, 1995). First we fix the probability of every base in the subsequence to some value $0 < X < 1$, and assign probabilities of the other bases according to $\frac{1-X}{4-1}$ (4 nucleic bases). Following Bailey and Elkan, we set X to 0.5. This gives us a set of seed probability matrices to be used as starting points for iterative improvement. Since motifs should occur in most sequences, we can select a random subset of the sequences and only generate candidate starting points from this subset.

4.2 Determining Possible Motif Occurrences

Rather than making the common assumption that each motif occurs only once per sequence, we allow for the possibility that a motif may occur multiple times in a single sequence. For each matrix and each sequence, we find the position that maximizes the match score. Now we set the threshold for deciding if a motif occurs at any position as the mean of match scores. Finally we add to the list of motif positions any position whose match score is greater than this threshold. This process defines a set of potential motif positions. Once these motif positions are defined, the seed probability matrices are no longer used.

4.3 Finding and Ranking Motif Candidates

After the likely motif positions are determined, DMS performs an iterative optimization procedure to find the motif probability matrix. Unlike current approaches that search all possible positions within a sequence, DMS only considers the potential motif po-

sitions determined in the previous step. This strategy significantly constrains the search space. For initialization, a randomly selected motif position from the potential positions in each sequence forms the initial probability matrix.

A sequence is then chosen at random for optimization. DMS optimizes the information content of the matrix by checking every potential motif position within the selected sequence. The position that gains the highest information content is chosen to update the matrix. The process is repeated until no improvement is noted. In each optimization cycle, the order of sequences is randomly shuffled. The randomization in each trial cycle is important to remove implicit biases, such as the order of the sequences, that can be harmful in search algorithms (Hampson and Kibler, 1996). At this point, in each sequence, the subsequence that contributes to the last updated matrix is determined. We then compute the mean of the match scores of the subsequences that form the matrix, and isolate all subsequences with a match score over the mean as possible motif repeats in each sequence. All these motif repeats in sequences are used to form the final motif matrix.

The same procedure is performed on all matrices to produce the motif candidates. Finally, DMS ranks the motif candidates according to its significance measure. Unlike other algorithms that use a probabilistic representation, DMS sets a threshold which defines whether or not a subsequence is a motif. This permits DMS to use the merit measure for ranking motifs. Other algorithms cannot directly apply the same merit measure.

A pseudocode description of matrix optimization procedure is given in Figure 2.

5 History/Related Work

We review some of the methods developed for the detection of the motifs. These methods were selected since they have been well-developed, are freely available over the internet, and represent a spectrum of different approaches.

CONSENSUS (Hertz *et. al.*, 1990; Hertz and Stormo, 1995) were pionneers in using computer search algorithms for identifying motifs. Their algorithms assume that there is exactly one occurrence per string and uses a beam search method with an information content evaluation function. The algorithms are somewhat limited in that they find a single motif (and variants) from a set of sequences and may lodge in local optima.

The Gibbs sampler (Lawrence *et. al.*, 1993) uses a probabilistic matrix to describe motifs, and adopts a search strategy based on random, iterative sampling. It is capable of finding multiple motifs in sequences when the number of occurrences of each motif in each sequence is known (Bailey & Elkan, 1995). It is computationally expensive and has difficulty learning short motifs.

The MEME algorithm (Bailey & Elkan, 1995) is a development of the Expectation Maximization (EM) algorithm introduced by Lawrence and Reilly, 1990. Similar to the Gibbs sampler, it uses a probabilistic representation. By repeatedly applying EM, MEME finds a matrix with maximum likelihood. MEME works particularly well on longer sequences. A potential limitation is that MEME performs less well on shorter patterns, as indicated by trials on real and artificial data.

Helden *et. al.* designed a simple, fast algorithm that detects over-represented oligonucleotides within sequences (Helden *et. al.*, 1998). This method exhaustively counts all oligonucleotide occurrences in the sequences, and estimates their statistical significance. This work highlighted the value of using multiplicity in identifying motifs with biological significance. However final identification of the motif is done manually using the program's output. To maintain the simplicity of the string representation, this approach sacrifices the expressiveness of probability matrices, making it less powerful in finding motifs with a large amount of variability.

We summarize the main design features of the various algorithms in Table 1.

6 Experiments on Real Domains

Recall that our goal is to describe the motif(s) that determines when a gene is expressed. From the literature, Helden *et. al.* defined ten families of genes that contain a number of known and probably unknown motifs. The known motifs for each family of yeast genes define ten learning tasks for evaluating the various motif algorithms. These families are listed in Table 2. Also recall that the (transcriptional) regulation of a yeast gene is primarily determined by motifs in the upstream region. As in the Helden study, we used the 800 bp upstream region of each ORF. These data are available from Saccharomyces Genome Database at Stanford[1].

We ran all the motif-finding algorithms above on these regulatory families except for the Helden algorithm, since his results were published. Except for DMS, none of the algorithms we tested provides any ranking information in its output. As they all adopt the matrix as the representation, and the matching threshold is implicit in the programs, our objective is to test whether they can identify the published motifs based on other

[1] http://genome-www.stanford.edu/Saccharomyces

```
Given: a set of biosequences, B
       A random subset of B, S
       the width of motif, W
Return: a set of ranked motif candidates, C

Step 1. Translation
        For each subsequence s in B Do
          Translate s into candidate probability matrix m via:
          m(i,base) = .50 if base occurs in position i
                    = .17 otherwise

Step 2. Determine possible motif positions
        For each sequence s in S Do
          Find highest match scoring subsequence in s
        Compute the mean of the highest match scores in S
        For each sequence s in S Do
          Set Potential Positions to those with match
          score >= mean

Step 3. Find and rank motif candidates
        Randomly choose a Potential Position in each sequence
        to initialize matrix M
        Repeat
          Randomly pick a sequence s in S
          Check if M's quality can be improved by using a
          different Potential Position in s
          Update matrix M
        Until no improvement in M's quality
        Compute the mean of match scores of subsequences
        contributing to M
        For each sequence s in S Do
          Isolate motif repeats to those with match score >= mean
        Form the final matrix FM with all repeats in S
        Put FM in C
Sort all motif candidates in C according to significance
Return C
```

Figure 2: Pseudocode of DMS

Table 1: Characteristics of Motif-Finding Algorithms

Algorithm	Search Strategy	Objective Function	Representation
CONSENSUS	beam search	information content	frequency matrix
Gibbs	stochastic hill-climbing	ratio of pattern probability to background probability	probabilistic matrix
MEME	EM variant	likelihood	probabilistic matrix
Helden	exhaustive	statistical significance assuming binomial distribution	base string
DMS	stochastic hill-climbing	information content and significance	probabilistic matrix

Table 2: Ten regulatory families and the associated published motifs

Family	Size	published motifs
NIT	7	GATAAG
MET	11	TCACGTG AAAACTGTGG
PHO	5	GCACGTGGG GCACGTTTT
PDR	7	TCCGCGGA
GAL	6	CGGNNNNNWNNNNNCCG
GCN	38	RRTGACTCTTT
INO	10	CATGTGAAWT
HAP	8	CCAAY
YAP	16	TTACTAA
TUP	25	KANWWWWATSYGGGGW

controllable parameters, e.g., the motif width and the number of motifs desired. Because of the variation in strategies of the algorithms, we allowed each algorithm to construct 100 motifs from each family. As biological experimentation is complex, expensive, and subject to noise, the literature typically only publishes the IUPAC code for regulatory motifs. Consequently we needed to construct a way to credit the algorithms that determined a probability matrix. Also the biological published motif contain some noise, We adopted the following procedure for determining a match. From each probability matrix we constructed a consensus pattern. If this consensus pattern matched the published motif in 80% of the positions of the motif, we counted this as a correct match. A base in the consensus sequence was allowed to match a disjunction of bases (as described by the IUPAC code) if the disjunction contained the base.

The experimental results are presented in Table 3. Column 2 to 5 shows whether the algorithm successfully identified the motifs. A "$*$" means the motifs(s) was successfully found, a "Δ" shows the motif(s) was contained in a longer pattern, and a blank indicates a failure.

In these trials, CONSENSUS did not find the GATAAG motif in the NIT family, which was reported by Helden *et. al.*. The algorithm requires setting several parameters whose influence is unclear. There may be some settings which permit the motif to be found. Moreover CONSENSUS failed to identify the published motifs in GCN, HAP, YAP and TUP regulatory families. Gibbs sampler found the published motifs in each family except the motifs in the HAP family, and the less conserved GCACGTTTT motif in PHO family. Gibbs is very sensitive to the setting of the expected number of motif occurrences. Wrong settings may hinder Gibbs sampler from isolating the correct motifs. MEME also identified all the published motifs except for the motifs in the HAP family, but it is also sensitive to whether to allow multiple appearances of a motif in any sequence or not. For example, allowing multiple appearances of a motif in any sequence prohibits MEME from detecting the target motif in the TUP family. In addition, MEME tended to detect longer elements even if we set it to find short motifs. Some of the shorter patterns are contained in longer ones, such as the motifs in the NIT family, the YAP and the MET. DMS identified all the published motifs in all regulatory families.

7 Experiments on Artificial Domains

The primary standard is how effectively these algorithms identify the reported motifs on real domains. However, as the biologists do not always have a complete idea of these regulatory families, and the collection of data sets is not extensive at the moment, it is useful to use synthetic domains to evaluate the various algorithms. While we have tried to maintain fidelity with real domains, we also had the ability to create motifs with known and controllable properties.

As the size of the families varied from 5 to 38 in the real domains, we used artificial families with sizes of 10 to 40 sequences. For the most part in real domains, the various algorithms did well at finding large motifs, but as the motif got shorter, the difficulty of finding them became higher. Consequently we created test sets with motif widths varying from 4 to 8 bases. The background sequences were generated either at random or by randomly shuffling real upstream regions from the yeast genome, e.g., the sets of 38 sequences are derived from the GCN family. To insert the motif into a sequence, we used four probabilities.

1. P_0 the probability of no artificial motif in a sequence
2. P_1 the probability of one artificial motif in a sequence,
3. P_2 the probability of two artificial motifs in a sequence,
4. P_B the probability of the preferred bases in the motif,

These fourteen artificial regulatory families are described in Table 4. The results are presented in Table 5, where we used the same test methodology as in the real domains. First, this data reinforces the conclusions from the experiments on real data, namely that CONSENSUS is unable to deal with variability in the motifs and that the stochastic search process of Gibbs only occasionally, but not always, lets it find the motif. MEME, which performed well on long motif patterns, failed to find the small seeded small motifs. On the other hand, DMS found all the seeded motifs.

8 Conclusions

Finding local consensus patterns in biosequences, i.e., motifs, is a very different problem than finding global alignments. We have reviewed the computational design of the leading approaches for finding motifs and provided the first empirical comparison of these on a common set of real and artificial problems. We have also introduced our own algorithm DMS for finding motifs, which combines many of the aspects of previous algorithms. This algorithm incorporated some novel constraints on the search that increases speed significantly without losing its ability to find motifs. On the chosen real domains, DMS and MEME performed nearly equivalently and much better than the

Table 3: Results of ten regulatory families

Family	CONSENSUS	Gibbs	MEME	DMS
NIT		*	Δ	*
MET	*	*	Δ	*
PHO	*	missed GCACGTTTT	*	*
PDR	*	*	*	*
GAL	*	*	*	*
GCN		*	*	*
INO	*	*	*	*
HAP	*			*
YAP		*	Δ	*
TUP		*	*	*

Table 4: 14 artificial regulatory families and the seeded motifs

	Family Size	Seq Length(bps)	Motif	P_0	P_1	P_2	P_B
1	10	800	CGCAA	0.0	0.8	0.2	1.0
2	10	800	CGTTT	0.0	0.8	0.2	1.0
3	38	800	CGCAA	0.0	0.8	0.2	1.0
4	38	800	CAGACA	0.0	0.8	0.2	1.0
5	38	800	CAGTC	0.0	0.8	0.2	0.9
6	38	800	CAGACA	0.0	0.8	0.2	0.9
7	38	800	CAGTCA	0.2	0.6	0.2	0.9
8	38	800	GTGTGTT	0.2	0.6	0.2	0.9
9	38	800	GCGAATT	0.2	0.6	0.2	0.9
10	38	800	CACGATA	0.2	0.6	0.2	0.9
11	40	500	CCCT	0.0	1.0	0.0	1.0
12	40	500	WCKGMCWG	0.0	1.0	0.0	1.0
13	40	500	WCTSACTG	0.0	1.0	0.0	0.9
14	40	500	WCTSACTG	0.0	1.0	0.0	0.8

Table 5: Results of 14 artificial families

Family	CONSENSUS	Gibbs	MEME	DMS
1		*		*
2		*		*
3				*
4		*	Δ	*
5				*
6		*		*
7				*
8				*
9				*
10				*
11				*
12		*	*	*
13		*	*	*
14				*

alternative algorithms. We believe that the DMS algorithm is superior at finding short motifs and that conclusion was supported by artificial experiments with seeded, variable, short motifs.

This research is part of a larger system that begins with collecting genes expression patterns using an Affymetrix gene-chip machine. Genes are then grouped into families with similar expression patterns via a new clustering algorithm. This affords us an automatic way to acquire families of similarly regulated genes. When DMS is run on these clusters, it has rediscovered know regulatory motifs and suggested additional motifs.

9 References

Bailey, T. and Elkan, C. (1995) "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization", Machine Learning, 21, p51-80.

Eddy, S. (1995) "Multiple Alignment using Hidden Markov Models", in Proceedings of International Conference on Intelligent Systems for Molecular Biology, p114-120.

Hampson, S. and Kibler, D. (1996) "Large Plateaus and Plateau Search in Boolean Satisfiability Problems: When to Give Up Searching and Start Again", In DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 26, p437-455.

Harr, R., Haggstrom, M. and Gustaffson, P. (1983) "Search Algorithm for Pattern Match Analysis of Nucleic Acid Sequences", Nucleic Acids Res., 11, p2943-2957.

Helden, J. V., Andre, B, and Collado-Vides, J. (1998) "Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies", Journal of Molecular Biology, 281, p827-842.

Hertz, G., Hartzell III, G. and Stormo, G. (1990) "Identification of Consensus Patterns in Unaligned DNA Sequences Known to be Functionally Related", Computer Applications in Biosciences, Vol 6, No 2, p81-92.

Hertz, G. and Stormo, G. (1995) "Identification of Consensus Patterns in Unaligned DNA and Protein Sequences: A Large-Deviation Statistical Basis for Penalizing Gaps", in Proceedings of the 3rd International Conference on Bioinformatics and Genome Research, p201-216.

Hughey, R. and Krogh, A. (1996) "Hidden Markov Models for Sequence Analysis: extension and analysis of the basic method", Computer Applications in Biosciences, Vol 12, No 2, p95-107.

Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A. and Wootton, J. (1993) "Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignments", SCIENCE, Vol 262, p208-214.

Lawrence, C. and Reilly, A. (1990) "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences", Protein: Structure Function and Genetics, 7, p41-51.

Staden, R. (1984) "Computer Methods to Locate Signals in Nucleic Acid Sequences", Nucleic Acids Res., 12, p505-519.

Stormo, G. (1988) "Computer Methods for Analyzing Sequence Recognition of Nucleic Acids", Annual Review of Biophysic and Biophysical Chemistry, 17, p241-263.