Detecting Motifs from Sequences

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

Abstract

The problem of multiple global comparison in families of biological sequences has been well-studied. Fewer algorithms have been developed for identifying local consensus patterns or motifs in biological sequence. These two important problems have different biological constraints and, consequently, different computational approaches. The difficulty of finding the biologically meaningful motifs results from (1) the variation among motif bases, (2) the alignment of motif position (sites) among the sequences, and (3) the multiplicity of motif occurrences within a given sequence. In this paper, we review and compare the main approaches for finding motifs. We also introduce our own approach, DMS, which combines two objective functions with an improved iterative sampling search method. We demonstrate the effectiveness of the various algorithms by comparing them on 10 real domains and 14 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 experimental analyses and interpretations. Among other genomes sequenced, yeast was completely sequenced in 1995. 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 might be guessed at 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 determined. 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 is the gene expressed ?

The first question has been heavily studied and primarily depends on characterizing a gene family. The most successful way of characterizing a gene has been based on probabilistic models, usually some instantiation of Hidden Markov Models (HMMs). HMMs work well for this problem since they provide a global model which allows insertions, deletions, and transpositions. 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 in animals is primarily determined by relatively short sequences in the upstream or surrounding region of a gene. These sequences vary in length from about 5 to 12, have large amount of variability in their base constituency, do not have inserts or deletes, do not occur in the same position, and sometimes occur multiple times. 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). These methods employ different representations, objective functions, and search strategies, and provide a basis for understanding the various approaches.

In addition, we present a new approach called DMS to detect motifs from sequences. DMS extends previous work by combining two objective functions with an improved iterative sampling technique. The performance of our algorithm and several others are compared on 10 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 14 artificial problems, which were designed to mimic real data.

2 Characteristics of Motif-finding Problem

Fundamentally the control of gene regulation is determined by chemical reactions which are, in turn, controlled by the shape and electrostatic charges of the molecules involved. Unfortunately this information is not 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.
- Insertion and deletions probably do not occur, as these would have a drastic effect on the local shape of the receptor.
- The pattern or motif should be common to most of these sequences and uncommon in all sequences. 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 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 functions are only heuristics. Once the objective function is determined, the goal is simply to find those patterns with high objective function value. Different objective functions 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, they are all heuristic. It is still unclear what is the best representation for patterns and the best objective function, as that relates to biology. More than likely, additional knowledge will need to be incorporated to get a better definition of a motif. 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 a hill-climbing strategy, but fell into local optimum. Standard approaches to repairing hill-climbing, such as beam search or adding a stochastic element was 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 a ranked order of motifs.

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, which contains 7 members, a possible 6-base motif matrix looks like the following. If we divide every element in the matrix by the total number of sequences, i.e., 7, we get a normalized matrix as shown in Figure 1.

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 two types of objective functions, the motif consensus quality and the motif significance. The consensus quality is used to guide the search for "common" motif candidates. Motif significance is used to rank motifs, estimating its biological significance.

We measure the consensus quality of a matrix by its relative information content. The information is calculated from two probabilities, the probability that each base (i.e., A, G, C, T) occurs in the genome, Pg_b, and the probability that each base occurs at each position in the motif, Pm_b. More precisely, the relative information content for a particular position n is given by:

$$I_n = \sum_{b \in A,G,C,T} Pm_b \lg \frac{Pm_b}{Pg_b}$$

The final consensus quality of a matrix is defined as the average quality over all positions.

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

$I = \frac{1}{W}\sum_{n=1}^{W} I_n$, where W is the width of the motif.

The significance measure is derived from the accuracy. It is simple and empirically effective. We define the significance of a motif b, by :

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

where $occ_S(b)$ is b's occurrences in S, and $occ_G(b)$ is b's occurrences in genome. A more complicated and computationally expensive measure of significance is given by Helden *et. al.* (Helden *et. al.*, 1998), but empirically we found the two measures give similar rankings.

Given a set of N biosequences, DMS carries out an iterative improvement search to find a matrix that maximizes the consensus quality defined above. This matrix is then a candidate for the motif. There are three main steps in DMS, which are described 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 other current approaches that search all possible positions within a sequence, DMS only considers the potential motif positions 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). When the process stops, 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 significance measure for ranking motifs, and the other algorithms cannot apply the same significance measure directly.

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) was one of the first motif-finding algorithms to identify matrices common to a set of sequences. The algorithm assumes that there is exactly one occurrence per string. The algorithm uses beam search with an information content evaluation function. The algorithm's limitations are that it only found a single motif (and variants) from a set of sequence and often got stuck in local optimum, missing the real motif.

The Gibbs sampler (Lawrence *et. al.*, 1993) uses a probabilistic matrix to describe a common pattern, and its search strategy is 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 an extension/variation on the Expectation Maximization (EM) algorithm introduced by Lawrence and Reilly, 1990. Like the Gibbs sampler, it also uses the probabilistic matrix as the representation. By repeatedly applying EM, MEME finds a matrix with maximum likelihood. In general MEME works very well, especially on longer sequences. A drawback is that MEME has difficulty finding short patterns, which is supported by experiments on real and artificial data.

Helden *et. al.* designed a simple 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 the multiplicity in identifying motifs, however it has a number of shortcomings. The motif is not really identified - this needs to be done manually from the output of the program. To maintain the simplicity of the representation, this approach sacrifices the expressiveness of probability matrices, making it unable to find motifs with a large amount of variability.

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

6 Experiments on Real Domains

Recall that our goal is to describe the motif that determines when a gene is expressed. From the literature, Helden *et. al.* defined ten families of genes that have known common regulatory site(s) or motif(s). Biologists believe that there are likely to be additional sites, but the known ones define ten learning tasks for evaluating the various algorithms. These families are described in more detail in Table 2. Also recall that the regulation of a gene is determined by motifs in the upstream region. We used the 800 bp upstream region for each gene, as this is the same sized region used by Helden in his experiments. The data is 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 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 the biological literature only publishes regulatory motifs in the IUPAC code, we needed to construct a way to credit the algorithms that determined a probability matrix. Also the biological published motif may, in fact, contain errors. We followed the following procedure for determining a match. From each probability matrix we constructed a consensus pattern. If this con-

[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

sensus 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.

CONSENSUS did not find the GATAAG motif in the NIT family as reported in Helden *et. al.'s* paper even though we specified the same matrix width and tried several different settings of the expected number of motif occurrences, including the one they used. There may be some other differences in the parameter settings. 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 perfect 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.

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		*	*	*

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 14 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. The surprising result is how often MEME failed to find the seeded small motifs. MEME found only three of the seeded motifs, and one of them is partially correct. 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. 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 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.

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				*

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.