Discovering Transcription Factor Binding Motif Sequences

I Lin

Department of Biology, Stanford University, CA, 94305

Introduction

In biology, sequence motifs are short sequence patterns, usually with fixed lengths, that represent many features of DNA, RNA, and protein molecules. Sequence motifs can represent transcription factor binding sites for DNA, splice junctions for RNA, and binding domains for proteins. Thus, discovering sequence motifs can lead to a better understanding of transcriptional regulation, mRNA splicing, and the formation of protein complexes. Furthermore, protein motifs can represent the active sites of enzymes or regions involved in protein structure and stability.

Motif discovery is an important computational problem because it allows the discovery of patterns in biological sequences in order to better understand the structure and function of the molecules the sequences represent. Especially, identifying regulatory elements, especially the binding sites in DNA for transcription factor, is important to understand the mechanisms that regulate gene expression. These DNA motif patterns are usually fairly short (5~20 base pairs long) and is known to recur in different genes or several times within a gene [1]. A DNA sequence can have zero, one, or multiple copies of a motif. In addition to these more common forms DNA motifs, there are also palindromic motifs (subsequence that is exactly the same as its own reverse complement) and gapped motifs (two smaller conserved sites separated by a gap) [2]. The high diversity and variability of motifs make them very difficult to identify.

A large number of algorithms for finding DNA motifs have been developed. These algorithms mostly detect overrepresented motifs and conserved motifs that might be good candidates for being transcription factor binding sites. Algorithms that detect overrepresented motifs deduce motifs by considering the regulatory region (promoter) of several co-regulated or co-expressed genes. Co-regulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, so their promoter regions might contain some common motifs that are binding sites for transcription factors. Thus, the way to detect these regulatory elements is to search for statistically overrepresented motifs in the promoter region of such a set of co-expressed genes. However, algorithms that detect overrepresented motifs perform not as well in higher organisms. To overcome this, some algorithms consider conserved motifs from orthologous species. Since selective pressure causes functional sequences to evolve slower than non-functional sequences, well-conserved sites represent possible candidates for DNA motifs. Recent algorithms have also combined the two approaches to achieve improvement in motif finding. In this report, we will review a few of the major recent developments in DNA motif finding algorithms.

General Techniques for Motif Discovery

Most motif finding algorithms fall into two major groups based on the combinatorial approach used: (1) word-based (string-based) method, represented by regular expressions (RE), or (2) probabilistic sequence models based on position weight matrices (PWM) [3]. The two methods have their own strengths and weaknesses.

The word-based method relies on exhaustively counting and comparing oligonucleotide frequencies, using regular expressions [4]. Regular expressions, often used in computer science, provide a concise and flexible means to "match" strings of text, which in this case are DNA sequence patterns. For example, a possible regular expression may be: "T-A-C-N(2,4)-G-T-A." This means that the RE matches any DNA sequences that begin with TAC and end with GTA, with a gap of length two to four in between that can be anything. The word-based method searches all possible regular expressions exhaustively to identify the REs whose match are most over-represented. The advantage of the word-based method is that it guarantees global optimum, since it does an exhaustive search. However, this also means that they are only suitable for short motifs. Also, although this method can be fast when implemented with suitable data structures, it is usually computationally expensive. This method is a good choice for finding motifs where all instances are identical. However, for typical transcription factor motifs that often have several weakly constrained positions, the word-based method can suffer [5].

The probabilistic approach involves representing the motif with a position weight matrix (PWM) [6]. A PWM defines the probability of each letter in the alphabet occurring at a specific position with an *n* by *m* matrix. *n* is the number of letters in the sequence (four for DNA) and *m* is the number of positions in the motif. The entry in row *i* and column *j* of the matrix is the probability of a letter *i* occurring at position *j* in the motif, represented by:

$$P_{i,j} \quad i \le n \quad j \le m$$

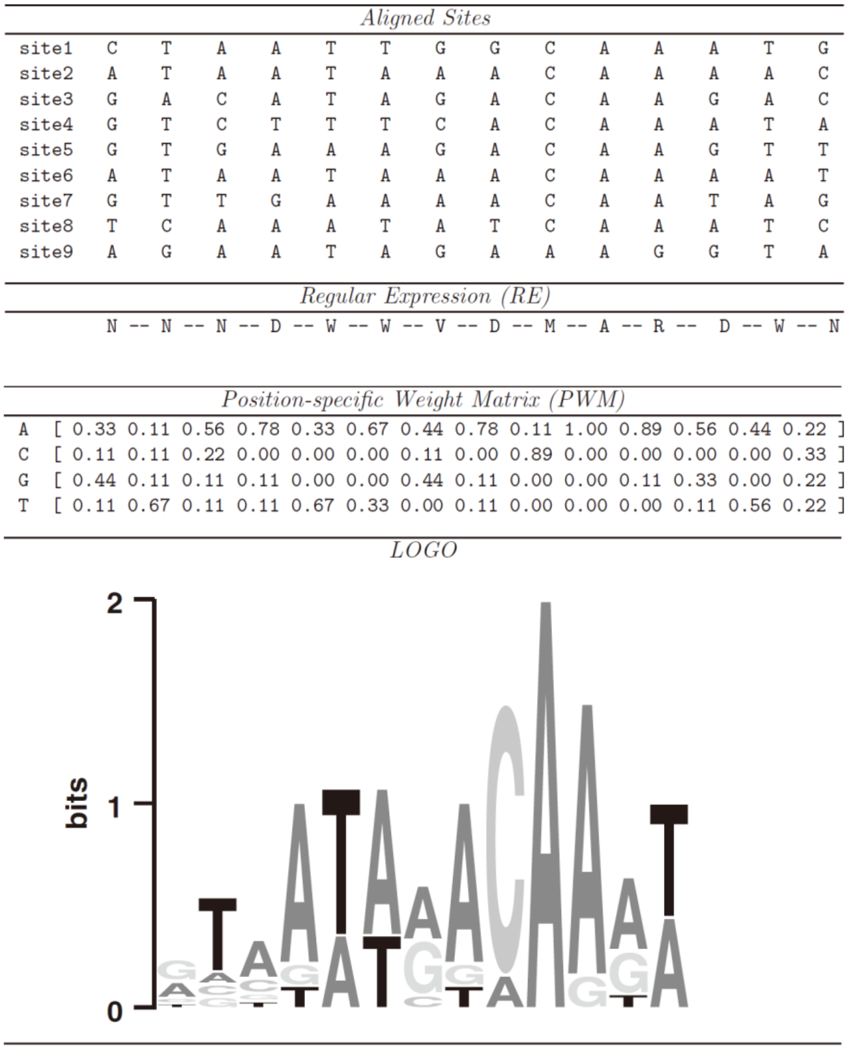
This model assumes that each position in the motif is statistically independent of the others. Thus, the probability of a sequence is just the product of the corresponding entries in the PWM. For example, the probability of the sequence "TACGTA" is just:

$$\Pr("TACGTA") = P_{T,1} \times P_{A,2} \times P_{C,3} \times P_{G,4} \times P_{T,5} \times P_{A,6}$$

The probabilistic approach searches the space of PWMs for motifs that maximize an objective function that is usually given by some sort of *log-likelihood* ratio (LLR):

$$LLR(PWM) = \sum_i \sum_j P_{i,j} \log_2 \frac{P_{i,j}}{f_i}$$

f_i is the overall probability of letter *i* in the sequence to be scanned for occurrences of the motif. The advantage for probabilistic approaches is that, compared with word-based methods, can have each letter "match" a particular motif position to varying degrees, rather than just match or no match. Many of the algorithms developed from probabilistic approaches are designed to find longer or more general motifs. However, these algorithms are not guaranteed to find globally optimal solutions, unlike word-based methods, since they employ some form of local search (such as Gibbs sampling, expectation maximization, or greedy algorithms) that may not converge to the global optimal solution.



Aligned Sites

site1	C	T	A	A	T	T	G	G	C	A	A	A	T	G
site2	A	T	A	A	T	A	A	A	C	A	A	A	A	C
site3	G	A	C	A	T	A	G	A	C	A	A	G	A	C
site4	G	T	C	T	T	T	C	A	C	A	A	A	T	A
site5	G	T	G	A	A	A	G	A	C	A	A	G	T	T
site6	A	T	A	A	T	A	A	A	C	A	A	A	A	T
site7	G	T	T	G	A	A	A	A	C	A	A	T	A	G
site8	T	C	A	A	A	T	A	T	C	A	A	A	T	C
site9	A	G	A	A	T	A	G	A	A	A	G	G	T	A

Regular Expression (RE)

N -- N -- N -- D -- W -- W -- V -- D -- M -- A -- R -- D -- W -- N

Position-specific Weight Matrix (PWM)

A	[	0.33	0.11	0.56	0.78	0.33	0.67	0.44	0.78	0.11	1.00	0.89	0.56	0.44	0.22	]
C	[	0.11	0.11	0.22	0.00	0.00	0.00	0.11	0.00	0.89	0.00	0.00	0.00	0.00	0.33	]
G	[	0.44	0.11	0.11	0.11	0.00	0.00	0.44	0.11	0.00	0.00	0.11	0.33	0.00	0.22	]
T	[	0.11	0.67	0.11	0.11	0.67	0.33	0.00	0.11	0.00	0.00	0.00	0.11	0.56	0.22	]

LOGO

Figure 1: The relationship between the motif sites, and RE, and a PWM. Nine example motifs are shown, with the corresponding RE and PWM, as well as the LOGO representation of the motif (Bailey, 2008).

Some literatures also categorize the motif finding algorithms into four classes based on the input sequences: (1) a "focused" approach: assemble a small set of sequences and search for over-represented patterns in the sequences, (2) a related "focused discriminative" approach: assemble two sets of sequences and look for patterns relatively over-represented in one of the input sets [7]. (3) a "phylogenetic" approach using sequence conservation information about the sequences in a single input set [8]. (4) a "whole-genome" approach looking for over-represented, conserved patterns in multiple alignments of the genomes of two or more species [9]. In this report, the “focused” approach of co-regulated genes will be emphasized

Word-based Algorithms

The motif finding algorithm Oligo-Analysis developed by van Helden *et al*. is based on the word-based approach [4]. This algorithm is conceptually very simple. It detects statistically significant motifs by counting the number of occurrences of each word or dyad and comparing these with expectation. Thus, this algorithm is exhaustive. However, it is limited to detecting only relatively simple patterns that include short motifs with highly conserved sequences. Later, van Helden extended this method to include spaced gap motifs (Dyad-Analysis) [10]. The major disadvantages of these algorithms are that no variations are allowed within an oligonucleotide.

Tompa *et al*. developed another algorithm with an exact word-based method to find short motifs in DNA sequences [11]. The algorithm takes into account both the absolute number of occurrences and the background distribution and creates a table that, for each length-*k* sequence *s*, records the number N_s of sequences containing an occurrence of *s*, where an occurrence allows for a small-fixed number *c* of substitution residues in *s*. Then, a reasonable measure of *s* as a motif would be based on how likely it is to have N_s occurrences if the sequences were drawn at random according to the background distribution. Now, let *X* be a single random sequence of the specified length *L*, with residues drawn randomly and independently from the background distribution. Suppose that p_s is the probability that *X* contains at least one occurrence of the length-*k* sequence *s*, allowing for *c* substitutions. We assume that *N* length-*L* random sequences of *X* are independent. Thus, the expected number of containing at least one occurrence of *s* among the *N* random sequence is:

$$Np_s$$

The standard deviation is:

$$\sqrt{Np_s(1-p_s)}$$

Thus, the *z-score* is:

$$M_s = \frac{N_s - Np_s}{\sqrt{Np_s(1-p_s)}}$$

The algorithm uses an exhaustive search to finds motifs with the greatest *z*-scores. Tompa *et al*. built upon this to develop the algorithm YMF (Yeast Motif Finder) to produce the motifs with greatest *z-scores*. This approach allowed variations within the oligonucleotide and made more accurate predictions.

Brazma *et al*. also used a word-based approach to develop a motif finding algorithm that looks for occurrences of regular expression-type patterns [12]. Many other algorithms were also developed that are similar to these approaches, combining word-based methods with graph-theory methods. For example, Sagot *et al*. introduced a word-based approach for motif finding that is based on the representation of a set of sequences with a suffix tree [13]. These implementations increased the computational efficiency of the algorithm, but have other drawbacks like more constraints on the target motifs to be searched.

Probabilistic Algorithms

Most probabilistic motif finding algorithms apply statistical techniques such as expectation maximization (EM) and Gibbs sampling algorithms. Gibbs sampling algorithms are used more extensively among the probabilistic approaches.

The EM algorithm iteratively maximizes the expected *log likelihood* using the Position Weight Matrix (PWM). The MEME algorithm developed by Bailey and Elkan is probably the most popular EM based algorithm for identifying motifs in unaligned sequences [14, 15]. MEME incorporated three ideas for discovering motifs: (1) subsequences that actually occur in the sequences are used as starting points for the EM algorithm to increase the probability of finding globally optimum motifs, (2) assumption that each sequence contains exactly one occurrence of the shared motif is removed, (3) 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. EM is a gradient descent method, so it cannot guarantee a global optimum. However, this means that the algorithm always converges in a predictable, relatively small number of iterations.

The Gibbs sampling method was originally developed by Lawrence *et al* [16]. The Gibbs sampler is a Markov Chain Monte Carlo (MCMC) approach for obtaining a sequence of radon samples from multivariate probability distribution. Markov chain means that the results from every step depend only on the results of the preceding one, like in EM. Monte Carlo means that the way to select the next step is not deterministic but rather based on random sampling. The MCMC also uses a probability matrix, and iterates until an optimal alignment is found when the ratio of motif probability to the background probability reaches a maximum.

More formally, we assume that we are given a set of N sequences $S_1,\ldots,S_N$, and we seek within each sequence mutually similar segments of specified width W. The algorithm proceeds through iterations of two steps, with each step maintaining an evolving data structure. The first is the “pattern description”, in the form of a probabilistic model of residue frequencies for each position i from 1 to W, and consisting of the variables $q_{i,1},\ldots,q_{i,4}$ indexed by W positions and 4 possible residues (for DNA). There is also a “background description”, $p_1,\ldots p_4$ with which residues occur in sites not described by the pattern. The second data structure is for alignment, which has a set of positions a_k for k from 1 to N, for the common pattern within the sequences. The algorithm proceeds through multiple iterations that execute the two steps: a predictive update step and a sampling step. For the predictive update step, one of the N sequences, z, is chosen at random. The pattern description $q_{i,j}$ and background frequencies p_j are then calculated from the current positions a_k in all sequences excluding z. For the sampling step, every possible segment of width W within sequence z is considered as possible instance of the pattern. The probabilities Q_x of generating each segment x according to the current pattern probabilities $q_{i,j}$ are calculated as well as the probabilities P_x of generating these segments by the background probabilities p_j. The weight $A_x = Q_x/P_x$ is assigned to segment x and a random segment is selected using these weights. Its position then becomes the new a_z. This iterative procedure allows accurate determination of motifs since the more accurate the pattern description constructed in step 1, the more accurate determination of its location in step 2, and this can converge to very accurate determination of motifs.

There are several methods based on Gibbs sampling approach. The following are three examples:

(1) Based on the Gibbs sampling approach, Roth *et al*. developed the motif finding algorithm AlignACE (Aligns Nucleic Acid Conserved Elements) [17]. This algorithm returns a series of motifs as weight matrices that are overrepresented in the input set of DNA sequences. AlignACE uses the MAP (maximum *a priori* log-likelihood) score to judge different motifs sampled, which gauges the degree of overrepresentation.

(2) Thijs *et al*. developed another motif finding algorithm, MotifSampler, which is also a modification of the Gibbs sampling algorithm [18]. MotifSampler uses a probability distribution to estimate the number of copies of the motif in a sequence and incorporates a higher-order Markov-chain background model.

(3) Another approach using the Gibbs sampling strategy, Liu *et al*. developed the motif finding algorithm, BioProspector [19]. BioProspector uses the promoter regions of co-regulated genes. It also uses zero to third-order Markov background models, and the significance of each motif is judged based on a motif score distribution estimated by a Monte Carlo method.

Other Algorithms and Applications

There are other algorithms, for example, that combine the word-based methods and probabilistic approaches, like the MDScan algorithm. The TAMO algorithm runs multiple motif discovery algorithms (MEME, AlignACE and MDscan) and combines the results [20]. Other approaches are based on other machine learning techniques, neural networks, and clustering algorithms. Algorithms based on phylogenetic foot-printing has the advantage of the co-regulated gene approach is that co-regulated methods require a way for identifying co-regulated genes; phylogenetic foot-printing approach is possible to identify motifs specific to even a single gene as long as they are sufficiently conserved across the many orthologous sequence considered. There are also algorithms that are based on promoter sequences of co-regulated genes and phylogenetic foot-printing.

These algorithms are often packaged into user-friendly interfaces, either on web servers or toolboxes. For example, the user-friendly interface, Toolbox of Motif Discovery (Tmod), integrates 12 widely used motif discovery programs: MDscan, BioProspector, AlignACE, Gibbs Motif Sampler, MEME, CONSENSUS, MotifRegressor, GLAM, MotifSampler, SeSiMCMC, Weeder and YMF [21].

Discussion

There are a large number of motifs finding algorithms are available that it is impossible to provide a comprehensive report. Each algorithm has its own advantages and disadvantages. The two main approaches for these motifs finding algorithms are word-based method and probabilistic approach. The main advantages of word-based method are that they are easy for us to visualize and for computers to search for the motif. It is also easier to compute the statistical significance of a motif divined as a regular expression. On the other hand, PWMs allow for a more flexible description of motifs because each letter can match a particular motif position to varying degree rather than simply matching or not matching. The main disadvantage of PWMs for motif discovery is that they are far more difficult for computer algorithms to search for. However, it is difficult to assess the performance and compare directly these algorithms. This is because each individual tool may do better on one type of data but do worse on other types of data. Also, since we still do not have a complete understanding of the biology of regulatory mechanism, it is difficult to evaluate the accuracy of these algorithms.

A few of the main algorithms, which are described above, are summarized below.

Table 1: Summary of the algorithms used for DNA motif finding

PWM-based algorithms	**Web Servers**
MEME	http://meme.nbcr.net
Gibbs	http://bayesweb.wadsworth.org/gibbs/gibbs.html
AlignACE	http://atlas.med.harvard.edu
MotifSampler	http://www.esat.kuleuven.ac.be/~dna/BioI/Software.html
BioProspector	http://seqmotifs.stanford.edu
MDScan	http://seqmotifs.stanford.edu

RE-based algorithms	**Web Servers**
Oligo/Dyad-Analysis	http://rsat.scmbb.ulb.ac.be/rsat/
YMF	http://wingless.cs.washington.edu/YMF
Weeder	http://www.pesolelab.it

Conclusions

Transcription factors bind to DNA motifs and modulate gene expression. Thus, identification of motifs in the promoter region of genes will help understand the regulation of gene expression. This problem has been of great interesting to computer scientists and biologists to use computational methods for motif finding. This provides a simple and efficient method to identify motifs without having to do time-consuming experiments.

There are myriads of algorithms available for motif finding, each with their advantage and disadvantages. Diverse approaches, including combinatorial enumeration, probabilistic modeling, mathematical programming, neural networks, and genetic algorithms, have been used. It is difficult to assess which motif finding tool is the best, since we do not have a clear understanding of the biology of regulatory mechanisms, so we lack an absolute standard against which to measure the correctness of these approaches. Thus, when using motif finding tools, it is important to use a few complementary tools in combination rather than relying on a single one.

Motif finding algorithms, combined with high-throughput transcriptional regulations microarray assays, will allow us to gain further understanding of transcription regulation and gene expression.

References:

1. Rombauts, S., et al., *PlantCARE, a plant cis-acting regulatory element database.* Nucleic Acids Res, 1999. **27**(1): p. 295-6.
2. Das, M.K. and H.K. Dai, *A survey of DNA motif finding algorithms.* BMC Bioinformatics, 2007. **8 Suppl 7**: p. S21.
3. Bailey, T.L., *Discovering sequence motifs.* Methods Mol Biol, 2008. **452**: p. 231-51.
4. van Helden, J., B. Andre, and J. Collado-Vides, *Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies.* J Mol Biol, 1998. **281**(5): p. 827-42.

5. Vilo, J., et al., *Mining for putative regulatory elements in the yeast genome using gene expression data.* Proc Int Conf Intell Syst Mol Biol, 2000. **8**: p. 384-94.

6. Bucher, P., *Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences.* J Mol Biol, 1990. **212**(4): p. 563-78.

7. Sinha, S., *Discriminative motifs.* J Comput Biol, 2003. **10**(3-4): p. 599-615.

8. Moses, A.M., D.Y. Chiang, and M.B. Eisen, *Phylogenetic motif detection by expectation-maximization on evolutionary mixtures.* Pac Symp Biocomput, 2004: p. 324-35.

9. Kellis, M., et al., *Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery.* J Comput Biol, 2004. **11**(2-3): p. 319-55.

10. van Helden, J., A.F. Rios, and J. Collado-Vides, *Discovering regulatory elements in non-coding sequences by analysis of spaced dyads.* Nucleic Acids Res, 2000. **28**(8): p. 1808-18.

11. Tompa, M., *An exact method for finding short motifs in sequences, with application to the ribosome binding site problem.* Proc Int Conf Intell Syst Mol Biol, 1999: p. 262-71.

12. Brazma, A., et al., *Predicting gene regulatory elements in silico on a genomic scale.* Genome Res, 1998. **8**(11): p. 1202-15.

13. Sagot, M.F., *Spelling approximate repeated or common motifs using a suffix tree.* Latin '98: Theoretical Informatics, 1998. **1380**: p. 374-390.

14. Bailey, T.L., et al., *MEME: discovering and analyzing DNA and protein sequence motifs.* Nucleic Acids Res, 2006. **34**(Web Server issue): p. W369-73.

15. Bailey, T.L., M.E. Baker, and C.P. Elkan, *An artificial intelligence approach to motif discovery in protein sequences: application to steriod dehydrogenases.* J Steroid Biochem Mol Biol, 1997. **62**(1): p. 29-44.

16. Lawrence, C.E., et al., *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment.* Science, 1993. **262**(5131): p. 208-14.

17. Roth, F.P., et al., *Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation.* Nat Biotechnol, 1998. **16**(10): p. 939-45.

18. Thijs, G., et al., *A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes.* J Comput Biol, 2002. **9**(2): p. 447-64.

19. Liu, X., D.L. Brutlag, and J.S. Liu, *BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes.* Pac Symp Biocomput, 2001: p. 127-38.

20. Gordon, D.B., et al., *TAMO: a flexible, object-oriented framework for analyzing transcriptional regulation using DNA-sequence motifs.* Bioinformatics, 2005. **21**(14): p. 3164-5.

21. Sun, H., et al., *Tmod: toolbox of motif discovery.* Bioinformatics, 2010. **26**(3): p. 405-7.