# How does DNA sequence motif discovery work?

Patrik D'haeseleer

**How can we computationally extract an unknown motif from a set of target sequences? What are the principles behind the major motif discovery algorithms? Which of these should we use, and how do we know we've found a 'real' motif?**

Extracting regulatory motifs[1] from DNA sequences seems to be all the rage these days. Take your favorite cluster of coexpressed genes, and with some luck you might hope to find a short pattern of nucleotides upstream of the transcription start sites of these genes, indicating a common transcription factor binding site responsible for their coordinate regulation. Easier said than done—the hunt for such a common motif may be like searching for the proverbial needle in a haystack. Consider the complexity of searching for imperfect copies of an unknown pattern, perhaps as small as 6–8 base pairs, occurring potentially thousands of bases upstream of some unknown subset of our genes of interest.

Let us first consider the problem in its most basic form. Given a set of sequences, which we have good reason to believe share a common binding motif, how do we go about extracting these often degenerate patterns from the set of sequences? Motif discovery algorithms subdivide into three distinct approaches: enumeration, deterministic optimization and probabilistic optimization.

## Enumeration

Enumerative algorithms exhaustively cover the space of all possible motifs, for a specific motif model description. For example, dictionary-based methods count the number of occurrences of all n-mers in the target sequences, and calculate which ones are most overrepresented. A motif description based on exact occurrence of specific words is too rigid for most real-world binding sites, but a number of similar overrepresented words may be combined into a more flexible motif description. Alternatively, one can search the space of all degenerate consensus sequences up to a given length, for example, using IUPAC codes for 2-nucleotide or 3-nucleotide degenerate positions in the motif[2]. Another enumerative approach describes a motif as a consensus sequence and an allowed number of mismatches, and uses an efficient suffix tree representation to find all such motifs in the target sequences[3]. Enumerative methods cover the entire search space, and therefore do not run the risk of getting stuck in a local optimum. On the other hand, the abstractions needed to achieve an enumerable search space may

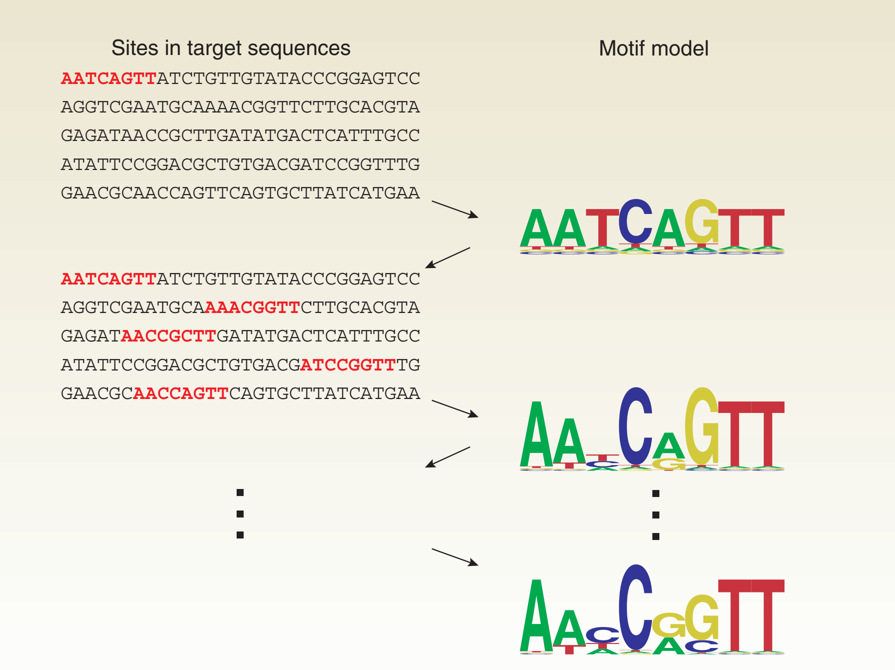
Sites in target sequences
Motif model
AATCAGTTATCTGTTGTATACCCGGAGTCC
AGGTCGAATGCAAAACGGTTCTTGCACGTA
GAGATAACCGCTTGATATGACTCATTTGCC
ATATTCCGGACGCTGTGACGATCCGGTTTG
GAACGCAACCAGTTCAGTGCTTATCATGAA
AATCAGTT
AATCAGTTATCTGTTGTATACCCGGAGTCC
AGGTCGAATGCAAAACGGTTCTTGCACGTA
GAGATAACCGCTTGATATGACTCATTTGCC
ATATTCCGGACGCTGTGACGATCCGGTTTG
GAACGCAACCAGTTCAGTGCTTATCATGAA
AACAGTT
AACGTT

**Figure 1** Starting from a single site, expectation maximization algorithms such as MEME[4] alternate between assigning sites to a motif (left) and updating the motif model (right). Note that only the best hit per sequence is shown here, although lesser hits in the same sequence can have an effect as well.

*Patrik D'haeseleer is in the Microbial Systems Division, Biosciences Directorate, Lawrence Livermore National Laboratory, 7000 East Ave., PO Box 808, L-448, Livermore, California 94551, USA.*
*e-mail: patrikd@llnl.gov*

overlook some of the subtle patterns present in real binding sites.

### Deterministic optimization

Expectation Maximization (EM) can be used to simultaneously optimize a position weight matrix (PWM) description of a motif[1], and the binding probabilities for its associated sites (**Fig. 1**). The weight matrix for the motif is initialized with a single n-mer subsequence, plus a small amount of background nucleotide frequencies. Next, for each n-mer in the target sequences, we calculate the probability that it was generated by the motif, rather than by the background sequence distribution. Expectation maximization then takes a weighted average across these probabilities to generate a more refined motif model. The algorithm iterates between calculating the probability of each site based on the current motif model, and calculating a new motif model based on the probabilities. It can be shown that this procedure performs a gradient descent, converging to a maximum of the log likelihood of the resulting model.

One popular implementation of the expectation maximization algorithm, MEME[4], performs a single iteration for each n-mer in the target sequences, selects the best motif from this set and then iterates only that one to convergence, avoiding local maxima. This partially enumerative nature of MEME provides some assurance that the algorithm is unlikely to get stuck in a poor local maximum. Additional motifs present in the set of target sequences can be found by masking the sequences matched by the first motif and rerunning the algorithm.

### Probabilistic optimization

Gibbs sampling can be viewed as a stochastic implementation of expectation maximization. Whereas the latter takes a weighted average across all subsequences (weighted with the current estimate of the probability that they belong to the motif), Gibbs sampling takes a weighted sample from these subsequences.

The motif model is typically initialized with a randomly selected set of sites, and every site in the target sequences is scored against this initial motif model. At each iteration, the algorithm probabilistically decides whether to add a new site and/or remove an old site from the motif model, weighted by the binding probability for those sites. The resulting motif model is then updated, and the binding probabilities recalculated. Given sufficient iterations, the algorithm will efficiently sample the joint probability distribution of motif models and sites assigned to the motif, focusing in on the best fitting combinations.

Interestingly, the term 'Gibbs sampling' was borrowed by analogy from statistical mechanics (via simulated annealing), where it refers to the inverse exponential relationship between the energy of a microstate and its probability, $p_m = Ce^{-\beta E_m}$. Because the weight matrix score of a site is actually related to the estimated binding energy of protein binding to that site[1], the analogy is particularly apt in this case.

### Which one?

Which methods should we use, out of this panoply of choices? In a recent large-scale comparison between 13 different motif discovery algorithms by Tompa *et al.*[5,6], enumerative approaches such as Weeder[3] and YMF[2] performed surprisingly well on a set of eukaryotic sequences with known motifs. (A complementary comparison on *Escherichia coli* sequences was performed by Hu, Li & Kihara[7].) However, each algorithm typically covers only a small subset of the known binding sites, with relatively little overlap between the algorithms. It is therefore advised to combine the results from multiple motif discovery tools, ideally covering a range of motif descriptions and search algorithms. For example, MotifSampler[8] (a Gibbs sampling implementation using a higher order Markov background model) was found to be complementary to a number of other, non-Gibbs, methods, including MEME[4], YMF[2] and Weeder[3].

Note that the implementation details of the algorithm—how the motifs are represented, whether the motif width and number of occurrences can be optimized, which objective function is being optimized—may be more important in practice than which optimization engine (EM or Gibbs sampling) is being used.

So let's say we've run a number of different algorithms, possibly returning multiple motifs from each. How do we decide which of these many motifs are biologically relevant? Nearly every algorithm uses a different evaluation criterion to optimize or score motifs, although often some variant of a log likelihood score is used:

### Information content, log likelihood and MAP score

Information content and relative entropy (see the earlier Primer on DNA sequence motifs[1]) measure how much a motif deviates from a background distribution of nucleotide frequencies, and can be read off at a glance from the sequence logo, by adding up the height of each stack of letters in the logo. However, it is based on a simplistic mononucleotide description of the background sequences, and does not account for the number of binding sites in the target sequences.

Likelihood of a model refers to the probability that the observed data could have been generated by the model in question. Typically, one optimizes the logarithm of this probability (hence 'log likelihood') with respect to the parameters of the model:

$$\log L(\mathrm{model} \mid \mathrm{data}) = \log \Pr(\mathrm{data} \mid \mathrm{model}) = \sum_i \log \Pr(\mathrm{data}_i \mid \mathrm{model})$$

Log likelihood allows for more sophisticated background models (typically 3rd or higher order Markov models), making it easier to rule out low-complexity repeats such as poly-A or poly-T sequences, which would otherwise show up as high information content motifs. It also takes the number of binding sites into account, not just how well they are conserved. In fact, the log likelihood is roughly proportional to the information content of the motif times the number of binding sites.

The maximum a posteriori probability (MAP) estimate of a model is the one that maximizes the probability of the model given the data. If all models are a priori equally likely, this is identical to the maximum likelihood estimate above. However, if we have some prior information regarding the motifs (for example, a total number of expected sites in the target sequences), we can use Bayes rule to bias the solution accordingly:

$$\Pr(\mathrm{model} \mid \mathrm{data}) = \frac{\Pr(\mathrm{data} \mid \mathrm{model})\Pr(\mathrm{model})}{\Pr(\mathrm{data})}$$

$$\log \Pr(\mathrm{model} \mid \mathrm{data}) \propto \log \Pr(\mathrm{data} \mid \mathrm{model}) + \log \Pr(\mathrm{model})$$

The MAP score therefore includes the log-likelihood score, plus it also allows incorporation of additional value judgments of the quality of the motif.

Although these various measures are widely used and have been very successful at discovering novel motifs, in practice none of them turns out to be sufficient to reliably distinguish known binding sites from spurious ones by themselves[6,7,9]. Therefore it is important not just to take the output of these algorithms at face value. (For some general guidelines on how to avoid or differentiate true from false-positive predictions, refer to **Box 1**).

### Other measures of motif quality

A number of other motif scores have proven useful to help distinguish 'real' from spurious motifs.

## Box 1 Practical guidelines

Given the rates of false positives and false negatives, any of these motif discovery tools should be used with caution, and their results should be examined carefully. Here are some useful guidelines for applying them effectively.

1. If possible, remove spurious patterns from the target sequences. For example, using RepeatMasker (http://www.repeatmasker.org/).
2. Use multiple motif prediction algorithms.
3. Run probabilistic algorithms multiple times—you may not get the best scoring motif on the first run.
4. If possible, ask for multiple motifs to be returned—the highest scoring one may not be the most biologically relevant.
5. If necessary, try a range of motif widths and expected number of sites (some tools will automatically optimize these parameters for you).
6. If needed, filter out motifs with biologically implausible distribution of information content (see the "block filtering" approach by Huber and Bulyk[9]).
7. Combine similar motifs, for example by calculating their similarity using AlignACE[10], clustering them, and taking the best representative from each cluster.
8. Use AlignACE[10] to match up with known motifs for the organism.
9. Evaluate the resulting motifs based on group specificity, set specificity, positional bias, etc.

Lately, a few packages have become available that combine multiple motif discovery algorithms, plus pre- and post-processing and analysis. Examples include MultiFinder[9] and RgS-Miner[12].

**Group specificity[10] (or site specificity[11]).** The probability of having this many target sequences containing the site (or this many sites within the target sequences), considering the prevalence of the motif throughout the genome.

**Sequence specificity[3,6].** Emphasizes both the number of sequences with binding sites, and the number of sites per sequence.

**Positional bias[10] or uniformity[6].** Measures how uniform the binding site locations are distributed, with respect to the transcription start site of the gene. Real transcription factor binding sites often (but not always) show a marked preference for a specific region upstream of the genes they regulate.

## Phylogenetic footprinting and ChIP-chip analysis

Although many of the current motif discovery algorithms were developed with the intent of looking for patterns in promoter regions of coexpressed genes, large-scale genome sequencing and recent innovative technologies in genomics are changing the landscape significantly.

In higher eukaryotes, transcription factor binding sites typically cluster together into *cis*-regulatory modules, which are under stronger evolutionary pressure than the surrounding sequence. The ready availability of genome data for closely related organisms allows us to use phylogenetic footprinting to focus the motif discovery algorithms on these blocks of conserved noncoding sequence, rather than on the vast intergenic wastelands, greatly increasing the sensitivity of the method.

The combination of chromatin immunoprecipitation with tiling microarrays is generating large sets of relatively short target sequences for known transcription factors of interest. As before, the size of the target sequences is bound to have a significant effect on the sensitivity of the method. This ChIP-chip approach is still fairly labor intensive, but, slowly but surely, the binding sites for most of the transcription factors of model organisms are being uncovered.

1. D'haeseleer. P. What are DNA sequence motifs? *Nat. Biotechnol.* **24**, 423–425 (2006).
2. Sinha, S. & Tompa, M. YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.* **31**, 3586–3588 (2003).
3. Pavesi, G. *et al.* Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res.* **32** (Web Server Issue), W199–W203 (2004).
4. Bailey, T.L. & Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2**, 28–36 (1994).
5. Tompa, M. *et al.* Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* **23**, 137–144 (2005).
6. Li, N. & Tompa, M. Analysis of computational approaches for motif discovery. *Alg. Mol. Biol.* **1**, 8 (2006).
7. Hu, J., Li, B. & Kihara, D. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Res.* **33**, 4899–4913 (2005).
8. Thijs, G. *et al.* A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comp. Biol.* **9**, 447–464 (2002).
9. Huber, B.R. & Bulyk, M.L. Meta-analysis discovery of tissue-specific DNA sequence motifs from mammalian gene expression data. *BMC Bioinformatics* **7**, 229 (2006).
10. Hughes, J.D. *et al.* Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae. J. Mol. Biol.* **296**, 1205–1214 (2000).
11. McGuire, A.M., Hughes, J.D. & Church, G.M. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res.* **10**, 744–757 (2000).
12. Huang, H.-D. *et al.* Identifying transcriptional regulatory sites in the human genome using an integrated system. *Nucleic Acids Res.* **32**, 1948–1956 (2004).