Chapter 12

Discovering Sequence Motifs

Timothy L. Bailey

Abstract

Sequence motif discovery algorithms are an important part of the computational biologist's toolkit. The purpose of motif discovery is to discover patterns in biopolymer (nucleotide or protein) sequences in order to better understand the structure and function of the molecules the sequences represent. This chapter provides an overview of the use of sequence motif discovery in biology and a general guide to the use of motif discovery algorithms. The chapter discusses the types of biological features that DNA and protein motifs can represent and their usefulness. It also defines what sequence motifs are, how they are represented, and general techniques for discovering them. The primary focus is on one aspect of motif discovery: discovering motifs in a set of unaligned DNA or protein sequences. Also presented are steps useful for checking the biological validity and investigating the function of sequence motifs using methods such as motif scanning—searching for matches to motifs in a given sequence or a database of sequences. A discussion of some limitations of motif discovery concludes the chapter.

Key words: Motif discovery, sequence motif, sequence pattern, protein domain, multiple alignment, position-specific scoring matrix, PSSM, position-specific weight matrix, PWM, transcription factor binding site, transcription factor, promoter, protein features.

1. Sequence Motifs and Biological Features

Biological sequence motifs are short, usually fixed-length, sequence patterns. Many features of DNA, RNA, and protein molecules can be well approximated by motifs. For example, sequence motifs can represent transcription factor binding sites (TFBSs), splice junctions, and binding domains in DNA, RNA, and protein molecules, respectively. Consequently, discovering sequence motifs can lead to a better understanding of transcriptional regulation, mRNA splicing and the formation of protein complexes.

Jonathan M. Keith (ed.), *Bioinformatics, Volume I: Data, Sequence Analysis, and Evolution, vol. 452*

Book doi: 10.1007/978-1-60327-159-2 Springerprotocols.com

Regulatory elements in DNA are among the most important biological features that are represented by sequence motifs. The DNA footprint of the binding sites for a transcription factor (TF) is often well described by a sequence motif. These TFBS motifs specify the order and nucleotide preference at each position in the binding sites for a particular TF. Discovering TFBS motifs and relating them to the TFs that bind to them is a key challenge in constructing a model of the regulatory network of the cell *(1, 2)*. Motif discovery algorithms have been used to identify many candidate TFBS motifs that were later validated by experimental methods.

Protein motifs can represent, among other things, the active sites of enzymes. They can also identify protein regions involved in determining protein structure and stability. The PROSITE, BLOCKS, and PRINTS databases *(3–5)* contain hundreds of protein motifs corresponding to enzyme active sites, binding sites, and protein family signatures. Motifs can also be used to identify features that confer particular chemical characteristics (e.g., thermal stability) on proteins *(6)*. Protein sequence motifs can also be used to classify proteins into families *(5)*.

The importance of motif discovery is born out by the growth in motif databases such as TRANSFAC, JASPAR, SCPD, DBTBS, and RegulonDB *(7–11)* for DNA motifs and PROSITE, BLOCKS, and PRINTS *(3–5)* for protein motifs. However, far more motifs remain to be discovered. For example, TFBS motifs are known for only about 500 vertebrate transcription factors TFs, but it is estimated that there are about 2,000 TFs in mammalian genomes alone *(6, 12)*.

Fixed-length motifs cannot represent all interesting patterns in biopolymer sequences. For instance, they are obviously not ideal for representing variable-length protein domains. For representing long, variable-length patterns, profiles *(13)* or HMMs *(14, 15)* are more appropriate. However, the dividing line between motifs and other sequence patterns (e.g., HMMs and profiles) is fuzzy, and is often erased completely in the literature. Some of the motif discovery algorithms discussed in the following sections, for example, do allow a single, variable-length "spacer," thus violating (slightly) our definition of motifs as being of fixed length. However, this chapter does not consider patterns that allow free insertions and deletions, even though these are sometimes referred to as motifs in the literature.

2. Representing Sequence Motifs

Biological sequence motifs are usually represented either as regular expressions (REs) or position weight matrices (PWMs). These two ways of describing motifs have different strengths and

weaknesses when it comes to expressive power, ease of discovery, and usefulness for scanning. Motif discovery algorithms exist that output their results in each of these types of motif representation. Some motif discovery algorithms do not output a description of the motif at all, but, rather, output a list of the "sites" (occurrences) of the motif in the input sequences. Any set of sites can easily be converted to a regular expression or to a PWM.

Regular expressions are a way to describe a sequence pattern by defining exactly what sequences of letters constitute a match. The simplest regular expression is just a string of letters. For example, "T-A-T-A-A-T" is a DNA regular expression that matches only one sequence: "TATAAT". (This chapter follows the PROSITE convention of separating the positions in an RE by a hyphen ("-") to distinguish them from sequences.) To allow more than one sequence to match an RE, extra letters (ambiguity codes) are added to the four-letter DNA sequence alphabet. For example, the IUPAC (16) code defines "W=A or T", so the RE "T-A-T-A-W-T" matches both "TATATT" and "TATAAT". For the 20-letter protein alphabet, ambiguity codes would be unwieldy, so sets of letters (enclosed in square brackets) may be included in an RE. Any of the letters within the square brackets is considered a match. As an added convenience, PROSITE protein motif REs allow a list of letters in curly braces, and any letter *except* the enclosed letters matches at that position. For example, the PROSITE N-glycosylation site motif is "N-{P}-[ST]-{P}". This RE matches any sequence starting with "N", followed by anything but "P", followed by an "S" or a "T", ending with anything but "P". As noted, some motif discovery programs allow for a variable-length spacer separating the two, fixed-length ends of the motif. This is particularly applicable to dyad motifs in DNA *(17, 18)*. The RE "T-A-C-N(2,4)-G-T-A" describes such a motif, in which "N" is the IUPAC "match anything" ambiguity code. The entry "-N(2,4)-" in the RE matches any DNA sequence of length from two to four, so sequences matching this RE have lengths from eight to ten, and begin and end with "TAC" and "GTA", respectively.

Whereas REs define the set of letters that may match at each position in the motif, PWMs define the *probability* of each letter in the alphabet occurring at that position. A PWM is an n by w matrix, where n is the number of letters in the sequence alphabet (four for DNA, 20 for protein), and w is the number of positions in the motif. The entry in row a, column i in the PWM, designated $P_{a,i}$, is the probability of letter a occurring at position i in the motif. Mathematically, PWMs specify the parameters of a position-specific multinomial sequence model that assumes each position in the motif is statistically independent of the others. A PWM defines a probability for every possible sequence of the correct width (w). The positional independence

assumption implies that the probability of a sequence is just the product of the corresponding entries in the PWM. For example, the probability of the sequence "TATAAT" according to a PWM (with six columns) is:

$$Pr(\text{"TATAAT"}) = P_{\text{T},1} \cdot P_{\text{A},2} \cdot P_{\text{T},3} \cdot P_{\text{A},4} \cdot P_{\text{A},5} \cdot P_{\text{T},6}.$$

As with REs, it is possible to extend the concept of PWMs to allow for variable-length spacers, but this is not commonly done by existing motif discovery algorithms.

For the purposes of motif scanning, many motif discovery algorithms also output a position-specific scoring matrix (PSSM), which is often confusingly referred to as a PWM. The entries in a PSSM are usually defined as:

$$S_{a,j} = \log_2 \frac{P_{a,j}}{f_a}, \qquad [1]$$

where f_a is the overall probability of letter a in the sequences to be scanned for occurrences of the motif. The PSSM score for a sequence is given by *summing* the appropriate entries in the PSSM, so the PSSM score of the sequence "TATAAT" is:

$$S(\text{"TATAAT"}) = S_{T,1} + S_{A,2} + S_{T,3} + S_{A,4} + S_{A,5} + S_{T,6}.$$

PSSM scores are more sensitive for scanning than probabilities because they take the "background" probability of different letters into account. This increases the match score for uncommon letters and decreases the score for common letters, thus reducing the rate of false-positives caused by non-uniform distribution of letters in sequences.

Underlying both REs and PWMs are the actual occurrences (sites) of the motif in the input sequences. The relationship among the motif sites, an RE and a PWM is illustrated in **Fig. 12.1**, which shows the JASPAR "*broad-complex 1*" motif. The nine motif sites from which this motif was constructed are shown aligned with each other at the top of the figure. The corresponding RE motif (using the IUPAC DNA ambiguity codes) is shown beneath the alignment. Below that, the counts of each letter in the corresponding alignment columns are shown. Below those, the corresponding PWM entries are shown. They were computed by normalizing each column in the counts matrix so that it sums to one. Beneath the PWM, the "LOGO" representation *(19)* for the motif is shown, where the height of each letter corresponds to its contribution to the motif's information content *(2)*.

Any alignment of motif sites can be converted into either an RE or PWM motif in the manner illustrated in **Fig. 12.1**. Usually a small amount (called a "pseudocount") is added to the counts in the position-specific count matrix before the PWM is

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 Count Matrix (PSCM)

A	[	3	1	5	7	3	6	4	7	1	9	8	5	4	2]
C	[	1	1	2	0	0	0	1	0	8	0	0	0	0	3]
G	[	4	1	1	1	0	0	4	1	0	0	1	3	0	2]
T	[	1	6	1	1	6	3	0	1	0	0	0	1	5	2]

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

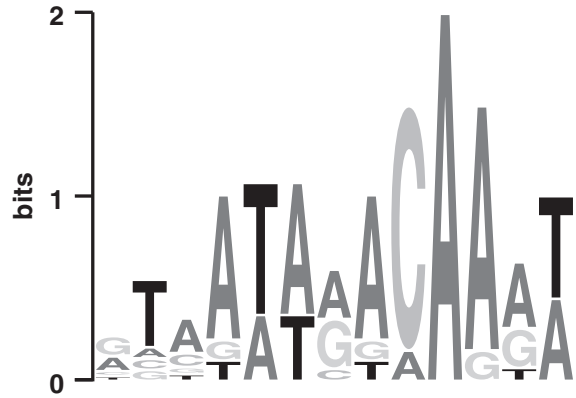

Fig. 12.1. Converting an alignment of sites into an RE and a PWM. The alignment of DNA sites is shown at the top. The RE (using the IUPAC ambiguity codes) is shown aligned below the sites. The corresponding counts of each letter in each alignment column—the position-specific count matrix (PSCM)—are shown in the next box. The PWM is shown below that. The last box shows the information content "LOGO" for the motif.

created by normalization in order to avoid probabilities of zero being assigned to letters that were not observed. This is sensible because, based on only a fraction of the actual sites, one cannot be certain that a particular letter *never* occurs in a real site.

Both PWMs and regular expressions are used by motif discovery algorithms because each has advantages. The main advantages of regular expressions are that they are easy for humans to visualize and for computers to search for. It is also easier to compute the statistical significance of a motif defined as a regular expression *(17, 20)*. On the other hand, PWMs allow for a more

nuanced description of motifs than regular expressions, because each letter can "match" a particular motif position to varying degrees, rather than simply matching or not matching. This makes PWM motifs (converted to PSSMs using [1]) more suitable for motif scanning than REs in most applications. When used to model binding sites in nucleotide molecules, there is evidence that PWMs capture some of the statistical mechanics of protein-nucleotide binding *(21–23)*. An extension of PWMs, called hidden Markov models (HMMs), has also been shown to be an invaluable way to represent protein domains (e.g., the PFAM database of protein domains) *(24)*. The main disadvantage of PWMs for motif discovery is that they are far more difficult for computer algorithms to search for. This is true precisely because PWMs are so much more expressive than REs.

3. General Techniques for Motif Discovery

Many approaches have been tried for *de novo* motif discovery. In general, they fall into four broad classes. The predominant approach can be called the "focused" approach: assemble a small set of sequences and search for over-represented patterns in the sequences relative to a background model. Numerous examples of available algorithms that use this approach are given in **Table 12.3**. A related approach can be called the "focused discriminative" approach: Assemble two sets of sequences and look for patterns relatively over-represented in one of the input sets *(25, 26)*. The "phylogenetic" approach uses sequence conservation information about the sequences in a single input set *(27–30)*. The "whole-genome" approach looks for over-represented, conserved patterns in multiple alignments of the genomes of two or more species *(31, 32)*. This chapter does not describe the "whole-genome" approach in any detail.

A sequence motif describes a pattern that recurs in biopolymer sequences. To be interesting to biologists, the pattern should correspond to some functional or structural feature that the underlying molecules have in common. None of the computational techniques for motif discovery listed in the preceding can guarantee to find only biologically relevant motifs. The most that can generally be said about a computationally discovered motif is that it is statistically significant, given underlying assumptions about the sequences in which it occurs.

The predominant approach to sequence motif discovery is the focused approach, which searches for novel motifs in a set of unaligned DNA or protein sequences suspected to contain a common motif. The next section discusses how the sequences

can be selected. RE-based motif discovery algorithms for the focused approach search the space of all possible regular expressions either exhaustively or heuristically (incompletely). Their objective is usually to identify the REs whose matches are most over-represented in the input sequences (relative to a background sequence model, randomly generated background sequences, or a set of negative control sequences). PWM-based motif discovery algorithms search the space of PWMs for motifs that maximize an objective function that is usually equal to (or related to) log likelihood ratio (LLR) of the PWM:

$$LLR(PWM) = \sum_{j=1}^{w} \sum_{a \in A} P_{a,j} \log_2 \frac{P_{a,j}}{f_a}, \quad [2]$$

where the $P_{a,j}$ are estimated from the predicted motif sites as illustrated in **Fig. 12.1**. The appropriateness of this objective function is justified by both Bayesian decision theory *(33)*, and, in the case of TFBSs, by binding energy considerations *(21, 23)*. When the background frequency model is uniform, LLR is equivalent to "information content".

4. Discovering Motifs in Sets of Unaligned Sequences

This section describes the steps necessary for successfully discovering motifs using the "focused" approach. Each motif discovery application is different, but most have the following steps in common:

1. Assemble: Select the target sequences.
2. Clean: Mask or remove "noise."
3. Discover: Run a motif discovery algorithm.
4. Evaluate: Investigate the validity and function of the motifs.

In the first step, you assemble a "dataset" of DNA or protein sequences that you believe may contain an unknown motif encoding functional, structural, or evolutionary information. Next, if appropriate, you mask or remove confounding sequence regions such as low-complexity regions and known repeat elements. You then run a motif discovery algorithm using your set of sequences and with parameter settings appropriate to your application. The next step is intended to weed out motifs that are likely to be chance artefacts rather than motifs corresponding to functional or structural features, and to try to glean more information about them. This step can involve determining if a discovered motif is similar to a known motif, or if its occurrences are conserved in orthologous genes. Each of these steps is described in more detail in the following sections.

4.1. Assemble: Select the Target Sequences

The most important step in motif discovery is to assemble a set of sequences that is likely to contain multiple occurrences of one or more motifs (*see* **Note 2**). For motif discovery algorithms to successfully discover motifs, it is important that the sequence set be as "enriched" as possible in the motifs. Obviously, if the sequences consist entirely of motif occurrences for a single motif, the problem of motif discovery is trivial (*see* **Fig. 12.1**). In practice, the guiding idea behind assembling a sequence set is to come as close as possible to such a set. To achieve this, all available background knowledge should be applied in order to:

- Include as many sequences as possible that contain the motifs.
- Keep the sequences as short as possible.
- Remove sequences that are unlikely to contain any motifs.

How you assemble your input sequence set depends, of course, on what type of motifs you are looking for and where you expect them to occur. In most applications, there are two basic steps:

1. Clustering
2. Extraction

First, you cluster genes (or other types of sequences) based on information about co-expression, co-binding, function, environment, or orthology to select ones likely to have a common motif. Second, you extract the relevant (portions of) sequences from an appropriate sequence database.

As an example, to discover regulatory elements in DNA, you might select upstream regions of genes that show co-expression in a microarray experiment *(34)*. Co-expression can be determined by clustering of expression profiles. Alternatively, you could use the sequences that bound to a TF in a ChIP-chip experiment *(1, 35)*. A third possibility is to use information on co-expressed promoters from CAGE tag experiments *(36, 37)*. To these sequence sets you might also add orthologous sequences from related organisms, the assumption being that the regulatory elements have been conserved in them.

To discover protein functional or structural sequence motifs, you could select proteins belonging to a given protein family based on sequence similarity, structure, annotation, or other means *(24, 38, 39)*. You might further refine the selection to only include proteins from organisms with a particular feature, such as the ability to live in extreme environments *(40)*. Another protein motif discovery application uses information from protein–protein interaction experiments. You can assemble a set of proteins that bind to a common host protein, in order to discover sequence motifs for the interacting domains.

Most algorithms require sets of sequences in FASTA format. Proteins are usually easily extracted directly from the available sequence databases. Genomic DNA is more problematic, since

annotation of genes, promoters, transcriptional start sites, introns, exons, and other important features is not always reliable. Several web servers available to aid you in extracting the relevant sequences for discovering regulatory elements in genomic DNA are shown in **Table 12.1**.

4.2. Clean: Mask or Remove "Noise"

Many genomic "phenomena" can masquerade as motifs and fool motif discovery algorithms (*see* **Note 3**). Things such as low-complexity DNA, low-complexity protein regions, tandem repeats, SINES, and ALUs all contain repetitive patterns that are problematic for existing motif-finding algorithms. It is therefore advisable to filter out these features from the sequences in the input set. This is done by running one or more of the programs described in **Table 12.2** on your set of sequences. Typically, the programs replace regions containing genomic "noise" with the ambiguity code for "match anything" in the appropriate sequence alphabet. This usually means "N" for DNA sequences and "X" for protein. Most motif discovery algorithms will not find motifs containing large numbers of these ambiguity codes, so they are effectively made invisible by this replacement process.

Table 12.2 lists some of the programs available to help you mask or remove confounding regions from your input sequence set. The DUST program *(41)* can be used to filter out low-complexity DNA. The XNU program *(42)* filters low-complexity (short period repeat) amino acid sequences. An alternative program for filtering out low-complexity protein

Table 12.1
Web servers for extracting upstream regions and other types of genomic sequence

Web server name	Function
RSA tools	Retrieve upstream regions for a large number of organisms. http://rsat.ulb.ac.be/rsat/
PromoSer	Retrieve human, rat, and mouse upstream regions, including alternative promoters. http://biowulf.bu.edu/zlab/PromoSer
UCSC genome browser *(74)*	View and extract genomic sequences and alignments of multiple genomes. http://genome.ucsc.edu

Table 12.2
Programs for filtering "noise" in DNA and protein sequences

Program name	Function
DUST	Filter low-complexity DNA. http://blast.wustl.edu/pub/dust
XNU	Filter low-complexity protein. http://blast.wustl.edu/pub/xnu
SEG	Filter low-complexity protein. http://blast.wustl.edu/pub/seg
RepeatMasker	Filter interspersed DNA repeats and low-complexity sequence. http://www.repeatmasker.org/cgi-bin/WEBRepeatMasker
Tandem Repeats Finder	Identify the positions of DNA tandem repeats. http://tandem.bu.edu/trf/trf.html

sequences is the SEG program *(43)*. Interspersed DNA repeats and low-complexity DNA sequence can both be filtered using the RepeatMasker program *(44)*. A web server is available for RepeatMasker, whereas at the time of this writing it was necessary to download, compile, and install the DUST, XNU, and SEG programs on your own computer. Tandem repeats can be identified in DNA using the "Tandem Repeats Finder" program. It has a web server that allows you to upload your sequence set (in FASTA format) for analysis. Of course, you should be aware that functional motifs can sometimes occur in the types of regions filtered by these programs, so caution is advised. It is important to study the documentation available with the programs to be sure you know what types of sequence they mask or identify. If you suspect that they may be masking regions containing your motifs of interest, you can always try running motif discovery algorithms on both the original and cleaned sets of sequences, and compare the results.

4.3. Discover: Run a Motif Discovery Algorithm

Many motif discovery algorithms are currently available. Most require installation of software on your computer. **Table 12.3** lists a variety of algorithms that have web servers in which you can upload your sequences directly, thus avoiding the need to install any new software. The table groups the algorithms according to whether they search for motifs expressed as REs or PWMs. Some of the algorithms are general purpose and can discover motifs in either DNA or protein sequences (MEME *(45)* and Gibbs *(46)*).

Table 12.3
Some motif discovery algorithms with web servers

PWM-Based algorithms

MEME	DNA or protein motifs using EM. http://meme.nbcr.net
Gibbs	DNA or protein motifs using Gibbs sampling. http://bayesweb.wadsworth.org/gibbs/gibbs.html
AlignACE	DNA motifs using Gibbs sampling. http://atlas.med.harvard.edu
CompareProspector	DNA motifs in eukaryotes using "biased" Gibbs sampling; requires multiple alignment. http://seqmotifs.stanford.edu
BioProspector	DNA motifs in prokaryotes and lower eukaryotes using Gibbs sampling. http://seqmotifs.stanford.edu
MDscan	DNA motifs; specialized for ChIP-chip probes. http://seqmotifs.stanford.edu

RE-Based algorithms

BlockMaker	Protein motifs. http://blocks.fhcrc.org/blocks/make_blocks.html
RSA tools	DNA motifs using RE-based or Gibbs sampler-based algorithms http://rsat.ulb.ac.be/rsat/
Weeder	DNA motifs using RE-based algorithm. http://www.pesolelab.it
YMF	DNA motifs using RE-based algorithm. http://wingless.cs.washington.edu/YMF

Combination algorithms

TAMO	Yeast, mouse, human; input as gene names or probe names, fetches upstream regions for you. http://fraenkel.mit.edu/webtamo

Some algorithms are specialized only for DNA (AlignACE *(47)*, BioProspector *(30)*, MDscan *(48)*, RSA Tools *(17, 49)*, Weeder *(50)*, and YMF *(51)*). CompareProspector *(52)* is specialized for DNA sequences and requires that you input your sequence set and conservation levels for each sequence position derived from a multiple alignment. BlockMaker *(53)* finds motifs only in protein sequences. The TAMO algorithm *(54)* runs multiple

motif discovery algorithms (MEME, AlignACE, and MDscan) and combines the results.

Many excellent algorithms are not included in **Table 12.3** because they did not appear to have a (working) web server at the time of this writing. Motif discovery algorithms require a great deal of computational power, so most authors have elected to distribute their algorithms rather than provide a web server. Other motif discovery algorithms include ANN-Spec *(26)*, Consensus *(55)*, GLAM *(56)*, Improbizer *(57)*, MITRA *(58)*, MotifSampler *(59)*, Phyme *(27)*, QuickScore *(60)*, and SeSiMCMC *(61)*.

Different classes of algorithms (RE- and PWM-based) have different strengths and weaknesses, so it is often helpful to run one or more motif discovery algorithms of each type on your sequence set. Doing this can increase the chances of finding subtle motifs. Also, the confidence in a given motif is increased when it is found by multiple algorithms, especially if the algorithms belong to different classes (*see* **Note 4**).

Some motif discovery algorithms (e.g., CompareProspector) can take direct advantage of conservation information in multiple alignments of orthologous sequence regions. This has been shown to improve the detection of TFBSs because they tend to be over-represented in sequence regions of high conservation *(62, 63)*. To find subtle motifs, it can also be useful to run each motif discovery algorithm with various settings of the relevant parameters. What the relevant parameters are depends on the particular problem at hand and the motif discovery algorithm you are using. You should read the documentation for the algorithm you are using for hints about what non-default parameter settings may be appropriate for different applications. In general, important parameters to vary include the limits on the width of the motif, the model used to model background (or "negative" sequences), the number of sites expected (or required) in each sequence, and the number of motifs to be reported (if the algorithm can detect multiple motifs).

4.4. Evaluate: Investigate the Validity and Function of the Motifs

One of the most difficult tasks in motif discovery is deciding which, if any, of the discovered motifs is "real." Three complementary approaches can aid you in this. First, you can attempt to determine whether a given motif is statistically significant. Second, you can investigate whether the function of the motif is already known or can be inferred. Third, you can look for corroborating evidence for the motif. Each of these approaches is discussed in the following.

Most motif discovery algorithms report motifs regardless of whether they are likely to be statistical artefacts. In other words, they "discover" motifs even in randomly generated (or randomly selected) sequences. This is sometimes referred to as the "GIGO"

rule: garbage-in, garbage-out. This, however, is not necessarily a bad thing; many truly functional DNA motifs are not statistically significant in the context of the kinds of sequence sets that can be assembled using clustered data from co-expression, ChIP-chip, CAGE, or other current technologies. So, it is important that motif discovery algorithms be able to detect these types of motifs even if they lie beneath the level of statistical significance that we might like. Measures of the statistical significance of a motif above the 0.05 significance level are still useful because they can be used to prioritize motifs for further validation.

Some motif discovery algorithms report an estimate of the statistical significance of the motifs they report. For example, MEME *(45)*, Consensus *(55)*, and GLAM *(56)* report the *E*-value of the motif: the probability of a motif of equal or greater information content occurring in a sequence set consisting of shuffled versions of each sequence. Motifs with very small (<0.05) *E*-values are statistically significant according to the given definition of random (shuffled sequences). The reported *E*-values are known to be conservative (too large), so motifs with *E*-values <0.05 may still be significant. Gibbs *(46)* uses a different statistical test (Wilcoxon signed-rank test) to determine motif significance. The relative merits of these two methods of assessing motif significance have not been studied.

Sometimes it is advisable to estimate motif significance empirically *(64)*. Many motif discovery algorithms do not make any attempt to report the statistical significance of the motifs they discover relative to the number of possible motifs that might have appeared in a randomly selected or generated sequence-set, so empirical estimation is the only available approach. Another reason to evaluate the significance of motifs empirically is that the motif significance estimates given by algorithms such as those named in the previous paragraph tend to be conservative, causing some biologically significant motifs to appear to be artefacts (*see* **Note 5**).

Empirical significance testing is very computationally expensive and therefore should generally be done using motif discovery algorithms installed on your local computer. Empirical significance testing is done by running the motif discovery algorithm hundreds of times on random sets of sequences of the same type and length, and with the same input parameters to the program, as were used in finding the motifs you are interested in evaluating. The motif scores for all the motifs found in the random runs are plotted as a histogram—the empirical score distribution. The significance of your real motifs' scores can be estimated by seeing where they lie on the histogram. The motif score can be either the information content score or the objective function score of the particular motif discovery method—usually some measure of over-representation. How you select (or

generate) the random sequence sets depends on your application. For example, if your real sequences are selected upstream regions of genes from a single organism, a reasonable random model would be to use randomly chosen upstream regions from the same organism.

Whether or not you choose to determine their statistical significance, you will probably want to determine as much as possible about the function of your motifs (*see* **Note 6**). To do this, you can use your motifs to search databases of motifs and motif families, and you can use your motifs individually and in groups to search databases of sequences for single matches and local clusters of matches. DNA motifs can be searched against known vertebrate TF motifs in JASPAR. The JASPAR database also contains motifs that represent the binding affinities of whole families of TFs. If your motif matches one of these family motifs, it may be the TFBS motif of a TF in that structural family. You can search your protein motif against the BLOCKS or PRINTS *(5)* database using the LAMA program *(65)* to identify if it corresponds to a known functional domain. These databases are summarized in **Table 12.4**.

You will also want to see if your motif occurs in sequences other than those in the sequence set in which it was discovered. This is done by scanning a database of sequences using your motif (or motifs) as the query. This can help validate the motif(s) and shed light on its (their) function. If the novel occurrences have a positional bias relative to some sequence landmark (e.g., the transcriptional start site), then this can be corroborating evidence that the motif may be functional *(47)*. In bacteria, real TFBSs are more likely to occur relatively close to the gene for their TF, so proximity to the TF can increase confidence in TFBSs predicted by motif scanning *(2)*. Similarly, when the occurrences of two or more motifs cluster together in several sequences, it may be evidence that the motifs are functionally

Table 12.4
Some searchable motif databases with web servers

Database	Description
JASPAR	Searchable database of vertebrate TF motifs and TF-family motifs. http://jaspar.genereg.net/
BLOCKS PRINTS	Databases of protein signatures. http://blocks.fhcrc.org/blocks-bin/LAMA_search.sh

related. (Care must be taken that the clustering of co-occurrences is not simply due to sequence homology.) The functions of the sequences in which novel motif occurrences are detected can also provide a hint to the motif's function. Scanning with multiple motifs can shed light on the interaction/co-occurrence of protein domains and on cis-regulatory modules (CRMs) in DNA.

Numerous programs are available to assist you in determining the location, co-occurrence and correlation with functional annotation of your motifs in other sequences. The MAST program *(66)* allows you to search a selection of sequence databases with one or more unordered protein or DNA motifs. The PATSER program *(55)* allows you to search sequences that you upload for occurrences of your DNA motif. Several tools are available for searching for cis-regulatory modules that include your TFBS motifs. They include MCAST *(67)*, Comet *(68)* and Cluster-buster *(69)*. To determine if the genomic positions of the matches to your motif or motifs are correlated with functional annotation in the GO (Genome Ontology) database *(70)*, you can use GONOME *(71)*. If the genomic positions are strongly correlated with a particular type of gene, this can shed light on the function of your motif. Some tools for motif scanning that are available for direct use via web servers are listed in **Table 12.5**.

Table 12.5
Some web servers for scanning sequences for occurrences of motifs

Program	Description
MAST	Search one or more motifs against a sequence database; provides a large number of sequence databases or allows you to upload a set of sequences. http://meme.nbcr.net
PATSER	Search a motif against sequences you upload. http://rsat.ulb.ac.be/rsat/patser_form.cgi
Comet, Clusterbuster	Search for cis-regulatory modules. http://zlab.bu.edu/zlab/gene.shtml
GONOME	Find correlations between occurrences of your motif and genome annotation in the GO database. http://gonome.imb.uq.edu.au/index.html

An important way to validate DNA motifs is to look at the conservation of the motif occurrences in both the original sequences and in sequences you scan as described in the previous paragraph. It has been shown that TFBSs exhibit higher conservation than the surrounding sequence in both yeast and mammals *(31, 32)*. Motifs whose sites (as determined by the motif discovery algorithm) and occurrences (as determined by scanning) show preferential conservation are less likely to be statistical artefacts. Databases such as the UCSC genome browser (*see* **Table 12.1**) can be consulted to determine the conservation of motif sites and occurrences.

5. Limitations of Motif Discovery

Awareness of the limitations of motif discovery can guide you to more success in the use of the approaches outlined in this chapter. Some limitations have to do with the difficulty of discovering weak motifs in the face of noise. Spurious motifs are another source of difficulty. Another limitation is caused by the difficulty in determining which sequences to include in the input sequence set (*see* **Note 1**).

You can often think of motif discovery as a "needle-in-a-haystack" problem where the motif is the "needle" and the sequences in which it is embedded is the "haystack." Because motif discovery algorithms depend on the relative over-representation of a motif in the input set of sequences, a motif is "weak" if it is not significantly over-represented in the input sequences relative to what is expected by chance (or relative to a negative set of sequences) *(72)*.

Over-representation is a function of several factors, including:

- The number of occurrences of the motif in the sequences
- How similar all the occurrences are to each other
- The length of the input sequences

The more occurrences of the motif the sequences contain, the easier they will be to discover. Therefore, adding sequences to the input set that have a high probability of containing a motif will increase the likelihood of discovering it. Conversely, it can be helpful to reduce the number of sequences by removing ones unlikely to contain motif occurrences. Many DNA motifs (e.g., TFBSs) tend to have low levels of similarity among occurrences, so it is especially important to limit sequence length and the number of "noise" sequences (ones not containing occurrences) in the input sequence set. Over-representation depends inversely on the length of the sequences, so it is always good to limit the

length of the input sequences as much as possible. Current motif discovery algorithms perform poorly at discovering TFBS when the sequences are longer than 1,000 bp.

Spurious motifs are motifs caused by non-functional, repetitive elements such as SINES, ALUs, and by skewed sequence composition in regions such as CpG islands. Such regions will contain patterns that are easily detected by motif discovery algorithms and may obscure real motifs. To help avoid this, you can pre-filter the sequences using the methods described in **Section 4.2**. In some cases, pre-filtering is not an option because the motifs of interest may lie in the regions that would be removed by filtering. For example, DNA regulatory elements often occur in or near CpG islands. In such cases, manual inspection using the methods of the previous section is necessary to remove spurious motifs. Using an organism-specific (or genomic-region–specific) random model is possible with some motif discovery algorithms, and may help to reduce the number of spurious motifs.

It is also important to be aware of the reliability of the methodologies used in selecting the input sequences for motif discovery. For example, sequences selected based on microarray expression data may miss many TFs because their level of expression is too low for modern methods to detect reliably *(2)*. ChIP-on-chip has become a popular procedure for studying genome-wide protein-DNA interactions and transcriptional regulation, but it can only map the probable protein-DNA interaction loci within 1-2Kbp resolution. Even if the input sequences all contain a TFBS motif, many TFBS motifs will not be detected in such long sequences using current motif discovery algorithms *(73)*. Another difficulty in discovering regulatory elements in DNA is that they can lie very far from the genes they regulate in eukaryotes, making sequence selection difficulty.

6. Notes

1. Be aware of the limitations of the motif discovery algorithms you use. For example, do not input an entire genome to most motif discovery algorithms—they are not designed for that and will just waste a lot of computer time without finding anything.
2. Use all available background information to select the sequences in which you will discover motifs. Include as many sequences as possible that contain the motifs. Keep the sequences as short as possible. Remove sequences that are unlikely to contain any motifs.

3. Prepare the input sequences carefully by masking or removing repetitive features that are not of interest to you such as ALUs, sines, and low-complexity regions. Filtering programs such as DUST, XNU, SEG, and RepeatMasker can help you do this.
4. Try more than one motif discovery algorithm on your data. They have different strengths and one program will often detect a motif missed by other programs.
5. Evaluate the statistical significance of your motifs. Remember that most motif discovery algorithms report motifs in any dataset, even though they may not be statistically significant. Even if the algorithm estimates the significance of the motifs it finds, these estimates tend to be very conservative, making it easy to reject biologically important motifs. So you should re-run the motif discovery algorithm on many sets of sequences that you select to be similar to your "real" sequences, but that you do not expect to be enriched in any particular motif. Compare the scores of your "real" motifs with those of motifs found in the "random" sequences to determine if they are statistically unusual.
6. Compare the motifs you discover to known motifs contained in appropriate motif databases such as those in **Table 12.4**.

References

1. Blais, A., Dynlacht, B. D. (2005) Constructing transcriptional regulatory networks. *Genes Dev* 19, 1499–1511.
2. Tan, K., McCue, L. A., Stormo, G. D. (2005) Making connections between novel transcription factors and their DNA motifs. *Genome Res* 15, 312–320.
3. Hulo, N., Bairoch, A., Bulliard, V., et al. (2006) The PROSITE database. *Nucleic Acids Res* 34, D227–D230.
4. Henikoff, J. G., Greene, E. A., Pietrokovski, S., et al. (2000) Increased coverage of protein families with the Blocks Database servers. *Nucleic Acids Res* 28, 228–230.
5. Attwood, T. K., Bradley, P., Flower, D. R., et al. (2003) PRINTS and its automatic supplement, prePRINTS. *Nucleic Acids Res* 31, 400–402.
6. La, D., Livesay, D. R. (2005) Predicting functional sites with an automated algorithm suitable for heterogeneous datasets. *BMC Bioinformatics* 6, 116.
7. Matys, V., Kel-Margoulis, O. V., Fricke, E., et al. (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* 34, D108–D110.
8. Sandelin, A., Alkema, W., Engstrom, P., et al. (2004) JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res* 32, D91–D94.
9. Zhu, J., Zhang, M. Q. (1999) SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics* 15, 607–611.
10. Makita, Y., Nakao, M., Ogasawara, N., et al. (2004) DBTBS: database of transcriptional regulation in Bacillus subtilis and its contribution to comparative genomics. *Nucleic Acids Res* 32, D75–D77.
11. Salgado, H., Gama-Castro, S., Peralta-Gil, M., et al. (2006) RegulonDB (version 5.0): *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res* 34(Database issue), D394–397.
12. Waterston, R. H., Lindblad-Toh, K., Birney, E., et al. (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature* 420, 520–562.

13. Gribskov, M., Veretnik, S. (1996) Identification of sequence pattern with profile analysis. *Methods Enzymol* 266, 198–212.
14. Eddy, S. R. (1998) Profile hidden Markov models. *Bioinformatics* 14, 755–763.
15. Krogh, A., Brown, M., Mian, I. S., et al. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol* 235, 1501–1531.
16. IUPAC-IUB Commission on Biochemical Nomenclature (1970) Abbreviations and symbols for nucleic acids, polynucleotides and their constituents. recommendations 1970. *Eur J Biochem* 15, 203–208.
17. van Helden, J., Andre, B., Collado-Vides, J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol* 281, 827–842.
18. van Helden, J., Rios, A. F., Collado-Vides, J. (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res* 28, 1808–1818.
19. Schneider, T. D., Stephens, R. M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res* 18, 6097–6100.
20. Reinert, G., Schbath, S., Waterman, M. S. (2000) Probabilistic and statistical properties of words: an overview. *J Comput Biol* 7, 1–46.
21. Schneider, T. D., Stormo, G. D., Gold, L., et al. (1986) Information content of binding sites on nucleotide sequences. *J Mol Biol* 188, 415–431.
22. Berg, O. G., von Hippel, P. H. (1987) Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *J Mol Biol* 193, 723–750.
23. Berg, O. G., von Hippel, P. H. (1988) Selection of DNA binding sites by regulatory proteins. II. The binding specificity of cyclic AMP receptor protein to recognition sites. *J Mol Biol* 200, 709–723.
24. Finn, R. D., Mistry, J., Schuster-Bockler, B., et al. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res* 34, D247–D251.
25. Sinha, S. (2003) Discriminative motifs. *J Comput Biol* 10, 599–615.
26. Workman, C. T., Stormo, G. D. (2000) ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. *Pac Symp Biocomput*, 467–478.
27. Sinha, S., Blanchette, M., Tompa, M. (2004) PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics* 5, 170.
28. Moses, A. M., Chiang, D. Y., Eisen, M. B. (2004) Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. *Pac Symp Biocomput* 324–335.
29. Siddharthan, R., Siggia, E. D., van Nimwegen, E. (2005) PhyloGibbs: a Gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol* 1, e67.
30. Liu, X., Brutlag, D. L., Liu, J. S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac Symp Biocomput*, 127–138.
31. Xie, X., Lu, J., Kulbokas, E. J., et al. (2005) Systematic discovery of regulatory motifs in human promoters and 3 UTRs by comparison of several mammals. *Nature* 434, 338-345.
32. Kellis, M., Patterson, N., Birren, B., et al. (2004) Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery. *J Comput Biol* 11, 319–355.
33. Duda, R. O., Hart, P. E. (1973) *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York.
34. Seki, M., Narusaka, M., Abe, H., et al. (2001) Monitoring the expression pattern of 1300 Arabidopsis genes under drought and cold stresses by using a full-length cDNA microarray. *Plant Cell* 13, 61–72.
35. Harbison, C. T., Gordon, D. B., Lee, T. I., et al. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature* 431, 99–104.
36. Kawaji, H., Kasukawa, T., Fukuda, S., et al. (2006) CAGE Basic/Analysis Databases: the CAGE resource for comprehensive promoter analysis. *Nucleic Acids Res* 34, D632–D636.
37. Kodzius, R., Matsumura, Y., Kasukawa, T., et al. (2004) Absolute expression values for mouse transcripts: re-annotation of the READ expression database by the use of CAGE and EST sequence tags. *FEBS Lett* 559, 22–26.
38. Tatusov, R. L., Fedorova, N. D., Jackson, J. D., et al. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 4, 41.
39. Andreeva, A., Howorth, D., Brenner, S. E., et al. (2004) SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res* 32, D226–D229.

40. La, D., Silver, M., Edgar, R. C., Livesay, D. R. (2003) Using motif-based methods in multiple genome analyses: a case study comparing orthologous mesophilic and thermophilic proteins. *Biochemistry* 42, 8988–8998.
41. Tatusov, R. L., Lipman, D. J. Dust, in the NCBI/Toolkit available at http://blast.wustl.edu/pub/dust/.
42. Claverie, J.-M., States, D. J. (1993) Information enhancement methods for large scale sequence analysis. *Comput Chem* 17, 191–201.
43. Wootton, J. C., Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases. *Methods Enzymol* 266, 554–571.
44. Smit, A., Hubley, R., Green, P. Repeatmasker, available at http://www.repeatmasker.org.
45. Bailey, T. L., Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol* 2, 28–36.
46. Thompson, W., Rouchka, E. C., Lawrence, C. E. (2003) Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic Acids Res* 31, 3580–3585.
47. Roth, F. P., Hughes, J. D., Estep, P. W., et al. (1998) Finding DNA regulatory motifs within unaligned non-coding sequences clustered by whole-genome mRNA quantitation. *Nat Biotechnol* 16, 939–945.
48. Liu, X. S., Brutlag, D. L., Liu, J. S. (2002) An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments. *Nat Biotechnol* 20, 835–839.
49. van Helden, J., Andre, B., Collado-Vides, J. (2000) A web site for the computational analysis of yeast regulatory sequences. *Yeast* 16, 177–187.
50. Pavesi, G., Mereghetti, P., Mauri, G., et al. (2004) Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res* 32, W199–W203.
51. Sinha, S., Tompa, M. (2003) YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res* 31, 3586–3588.
52. Liu, Y., Liu, X. S., Wei, L., Altman, R. B., et al. (2004) Eukaryotic regulatory element conservation analysis and identification using comparative genomics. *Genome Res* 14, 451–458.
53. Henikoff, S., Henikoff, J. G., Alford, W. J., et al. (1995) Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene* 163, GC17–GC26.
54. Gordon, D. B., Nekludova, L., McCallum, S., et al. (2005) TAMO: a flexible, object-oriented framework for analyzing transcriptional regulation using DNA-sequence motifs. *Bioinformatics* 21, 3164–3165.
55. Hertz, G. Z., Stormo, G. D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* 15, 563–577.
56. Frith, M. C., Hansen, U., Spouge, J. L., et al. (2004) Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res* 32, 189–200.
57. Ao, W., Gaudet, J., Kent, W. J., et al. (2004) Environmentally induced foregut remodeling by PHA4/FoxA and DAF-12/NHR. *Science* 305, 1742–1746.
58. Eskin, E., Pevzner, P. A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics* 18, S354–S363.
59. Thijs, G., Marchal, K., Lescot, M., et al. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J Comput Biol* 9, 447–464.
60. Regnier, M., Denise, A. (2004) Rare events and conditional events on random strings. *Discrete Math Theor Comput Sci* 6, 191–214.
61. Favorov, A. V., Gelfand, M. S., Gerasimova, A. V., et al. (2005) A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics* 21, 2240–2245.
62. Tagle, D. A., Koop, B. F., Goodman, M., et al. (1988) Embryonic epsilon and gamma globin genes of a prosimian primate *(Galago crassi caudatus)*. Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J Mol Biol* 203, 439–455.
63. Duret, L., Bucher, P. (1997) Searching for regulatory elements in human noncoding sequences. *Curr Opin Struct Biol* 7, 399–406.
64. Macisaac, K. D., Gordon, D. B., Nekludova, L., et al. (2006) A hypothesis-based approach for identifying the binding specificity of regulatory proteins from chromatin immunoprecipitation data. *Bioinformatics* 22, 423–429.

65. Pietrokovski, S. (1996) Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Res* 24, 3836–3845.
66. Bailey, T. L., Gribskov, M. (1998) Combining evidence using p-values: application to sequence homology searches. *Bioinformatics* 14, 48–54.
67. Bailey, T. L., Noble, W. S. (2003) Searching for statistically significant regulatory modules. *Bioinformatics* 19, II16–II25.
68. Frith, M. C., Spouge, J. L., Hansen, U., et al. (2002) Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res* 30, 3214–3224.
69. Frith, M. C., Li, M. C., Weng, Z. (2003) Cluster-Buster: finding dense clusters of motifs in DNA sequences. *Nucleic Acids Res* 31, 3666–3668.
70. Ashburner, M., Ball, C. A., Blake, J. A., et al. (2000) Gene ontology: tool for the unification of biology. *Nat Genet* 25, 25–29.
71. Stanley, S., Bailey, T., Mattick, J. (2006) GONOME: measuring correlations between gene ontology terms and genomic positions. *BMC Bioinformatics* 7, 94.
72. Keich, U., Pevzner, P. A. (2002) Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics* 18, 1382–1390.
73. Tompa, M., Li, N., Bailey, T. L., et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol* 23, 137–144.
74. Kent, W. J., Sugnet, C. W., Furey, T. S., et al. (2002) The human genome browser at UCSC. *Genome Res* 12, 996–1006.