BRIEFINGS IN BIOINFORMATICS. VOL 10. NO 5. 525–536
Advance Access publication June 24, 2009
doi:10.1093/bib/bbp032

Finding sequence motifs in prokaryotic genomes—a brief practical guide for a microbiologist

Jan Mrázek

Submitted: 13th March 2009; Received (in revised form): 3rd June 2009

Abstract

Finding significant nucleotide sequence motifs in prokaryotic genomes can be divided into three types of tasks: (1) supervised motif finding, where a sample of motif sequences is used to find other similar sequences in genomes; (2) unsupervised motif finding, which typically relates to the task of finding regulatory motifs and protein binding sites and (3) exploratory motif finding, which aims to identify potential functionally significant sequence motifs as those that are unusual in some statistical sense. This article provides a conceptual overview for each type of task, a brief description of basic algorithms used in their solution, and a review of selected relevant software available online.

Keywords: *supervised motif finding; unsupervised motif finding; phylogenetic footprinting; r-scan statistics; protein binding sites; software review*

INTRODUCTION

Accelerated sequencing of prokaryotic genomes in the past decade has been accompanied by a rise in the use of bioinformatic tools by microbiologists. It would be difficult to find a microbiologist today who has never accessed online biological databases, searched for sequences using BLAST, aligned nucleotide or protein sequences, or built a phylogenetic tree. Bioinformaticians and software developers have facilitated this process by increased emphasis on convenience and ease of use of software and databases, which in turn allowed microbiologists to use increasingly sophisticated techniques with minimal effort. As a computational biologist and a developer, as well as user, of bioinformatic tools who recently settled in a microbiology department, I had an opportunity to approach the field of bioinformatics from both sides—the 'bio' and the 'informatics'. My colleagues and their students have often asked me for help in solving practical computational issues relevant to their work. Many of their questions relate to finding sequence motifs in prokaryotic genomes. Here I review some of the tools and approaches for computational analyses of prokaryotic genomes, concentrating on various forms of motif finding in genomic nucleotide sequences and related tasks. Examples are provided demonstrating the importance of choosing the right method for the specific task or question at hand.

SUPERVISED MOTIF FINDING

Conceptual overview

The most common application of supervised motif finding starts with a set of short nucleotide sequences that are known to have a particular function (for example, binding sites for a specific protein) and the task at hand is to find all similar sequences in a genome. The key questions in solving this task include definition of what constitutes a 'similar' sequence and a suitable formal representation of the aligned motif sequences. A commonly used technique is to determine a consensus sequence and consider every sequence matching the consensus to be 'similar'. The simplicity of consensus sequence representation of the motif, its intuitive interpretation,

Corresponding author. Department of Microbiology and Institute of Bioinformatics, University of Georgia, Athens, GA 30602-2605, USA. Tel: +1 706-542-1065; Fax: +1 706-542-2674; E-mail: mrazek@uga.edu

Jan Mrázek received his PhD in Biophysics in the Czech Republic. He was a postdoctoral fellow in the Department of Mathematics at Stanford University (laboratory of Samuel Karlin) and he is currently in the Department of Microbiology at the University of Georgia.

and availability of many programs to find motifs matching a particular consensus (see below) makes it an attractive and frequently utilized choice. While not particularly sophisticated, the consensus representation will serve well for motifs that are highly conserved. However, for less conserved sequence motifs, the consensus representation is too rigid. Position-specific score matrix (PSSM) is a more flexible alternative to representing sequence motifs. In this approach, every nucleotide a (A, C, G or T) is assigned a numerical score $s_{a,i}$ for each position i in the motif, such that nucleotides frequently occurring at that particular position have higher scores than the nucleotides that are rarely found at that position. Then any nucleotide sequence of the same length n as the motif can be assigned a score

$$S = \sum_{i=1}^{n} s_{a_i,i}, \qquad (1)$$

where a_i is the nucleotide at position i in the sequence at hand. Any sequence scoring above some cutoff S_0 (see below) can now be considered a match to the motif. To construct a PSSM for a particular motif, we start with the alignment of the known (e.g. experimentally determined) motif sequences (Step 1, Table 1). The second step involves converting the alignment into the frequency matrix $\{N_{a,i}\}$ (Table 2), where $N_{a,i}$ is the number of sequences in the alignment that have the nucleotide a at the position i. Next (let us call it step 2½), we eliminate zeroes from the frequency matrix. The importance of this step will be apparent later. This is typically done via pseudocounts—adding a small number to each value in the matrix. There are no universal rules for determining the most appropriate pseudocounts and the selection of pseudocount values is usually purely empirical (e.g. [1]). In this example, we shall use ¼ as the pseudocount value and convert the frequency matrix $\{N_{a,i}\}$ into an adjusted frequency matrix $\{N'_{a,i}\}$, where $N'_{a,i} = N_{a,i} + 1/4$. In step 3, we convert the adjusted frequency matrix into a probability matrix $\{p_{a,i}\}$, where $p_{a,i} = N'_{a,i} / \sum_{a=A,C,G,T} N'_{a,i}$ (Table 3); in our example $\sum_{a=A,C,G,T} N'_{a,i} = 21$. The values $p_{a,i}$ represent estimated probabilities that a Crp binding site has the nucleotide a at position i, and the matrix represents a probabilistic model of the Crp binding site. Note that due to the pseudocounts, all probabilities are higher than zero. If the scores are defined as $s_{a,i} = \ln p_{a,i}$, then using the formula (1) above one obtains $e^S = \prod_{i=1}^{n} p_{a_i,i}$ (product of probabilities $p_{a_i,i}$), which can be interpreted as

Table 1: Construction of PSSM, step 1: aligned *E. coli* Crp binding sites

Sequence ID	Sequence
#1	ATTCGTGATAGCTGTCGTAAAG
#2	TTTTGTTACCTGCCTCTAACTT
#3	AAGTGTGACGCCGTGCAAATAA
#4	TGCCGTGATTATAGACACTTTT
#5	ATTTGCGATGCGTCGCGCATTT
#6	TAATGAGATTCAGATCACATAT
#7	TAATGTGACGTCCTTTGCATAC
#8	GAAGGCGACCTGGGTCATGCTG
#9	AGGTGTTAAATTGATCACGTTT
#10	CGATGCGAGGCGGATCGAAAAA
#11	AAATTCAATATTCATCACACTT
#12	AGATGTGAGCCAGCTCACCATA
#13	AGATGTGATTAGATTATTATTC
#14	AATTGTGATGTGTATCGAAGTG
#15	TTATTTGAACCAGATCGCATTA
#16	AAATGTAAGCTGTGCCACGTTT
#17	AAGTGTGACATGGAATAAATTA
#18	TTGTTTGATTTCGCGCATATTC
#19	AAACGTGATTTCATGCGTCATT
#20	ATGTGTGCGGCAATTCACATTT

Sequences were obtained from the DPInteract database (http://arep.med.harvard.edu/dpinteract/ [52]). Only the first 20 of the 49 Crp binding sites listed in DPInteract are displayed.

Table 2: Construction of PSSM, step 2: frequency matrix $\{N_{a,i}\}$ derived from the alignment in Table 1

Position (i)	Nucleotide (a)			
	A	C	G	T
1	12	1	1	6
2	9	0	5	6
3	10	1	5	4
4	0	3	1	16
5	0	0	17	3
6	1	4	0	15
7	2	0	16	2
8	19	1	0	0
9	2	5	4	9
10	4	5	6	5
11	2	7	1	10
12	4	5	8	3
13	4	3	9	4
14	7	4	4	5
15	2	1	4	13
16	1	17	0	2
17	11	0	7	2
18	5	10	0	5
19	14	2	3	1
20	4	3	1	12
21	5	0	0	15
22	5	3	3	9

The frequency matrix consists of counts $N_{a,i}$ of the base a at the alignment position i.

Table 3: Construction of PSSM, step 3: probability matrix $\{p_{a,i}\}$ derived from the frequency matrix in Table 2 using pseudocounts equal to 0.25

Position (i)	Nucleotide (a)			
	A	C	G	T
1	0.58	0.06	0.06	0.30
2	0.44	0.01	0.25	0.30
3	0.49	0.06	0.25	0.20
4	0.01	0.15	0.06	0.77
5	0.01	0.01	0.82	0.15
6	0.06	0.20	0.01	0.73
7	0.11	0.01	0.77	0.11
8	0.92	0.06	0.01	0.01
9	0.11	0.25	0.20	0.44
10	0.20	0.25	0.30	0.25
11	0.11	0.35	0.06	0.49
12	0.20	0.25	0.39	0.15
13	0.20	0.15	0.44	0.20
14	0.35	0.20	0.20	0.25
15	0.11	0.06	0.20	0.63
16	0.06	0.82	0.01	0.11
17	0.54	0.01	0.35	0.11
18	0.25	0.49	0.01	0.25
19	0.68	0.11	0.15	0.06
20	0.20	0.15	0.06	0.58
21	0.25	0.01	0.01	0.73
22	0.25	0.15	0.15	0.44

Table 4: Construction of PSSM, step 3: the final PSSM

Position (i)	Nucleotide (a)			
	A	C	G	T
1	0.86	−1.45	−1.45	0.19
2	0.58	−3.06	−0.02	0.19
3	0.69	−1.45	−0.02	−0.20
4	−3.03	−0.50	−1.45	1.15
5	−3.03	−3.06	1.17	−0.46
6	−1.42	−0.23	−3.06	1.08
7	−0.83	−3.06	1.11	−0.83
8	1.32	−1.45	−3.06	−3.03
9	−0.83	−0.02	−0.23	0.58
10	−0.20	−0.02	0.16	0.02
11	−0.83	0.31	−1.45	0.69
12	−0.20	−0.02	0.44	−0.46
13	−0.20	−0.50	0.55	−0.20
14	0.34	−0.23	−0.23	0.02
15	−0.83	−1.45	−0.23	0.94
16	−1.42	1.17	−3.06	−0.83
17	0.78	−3.06	0.31	−0.83
18	0.02	0.65	−3.06	0.02
19	1.01	−0.86	−0.50	−1.42
20	−0.20	−0.50	−1.45	0.86
21	0.02	−3.06	−3.06	1.08
22	0.02	−0.50	−0.50	0.58

The log-odds ratio scores (Formula 2) for the sequence motifs in Table 1.

the probability that the sequence at hand belongs to the same class as the sequences in the initial alignment.

One could stop here and use the scores $s_{a,i} = \ln p_{a,i}$. But consider a sequence motif in a genome with a very low G + C content—should not positions with a conserved G or C be assigned higher significance than positions with conserved A or T? To account for differences in genomic backgrounds we add a probabilistic model for the background, that is, a probability that a sequence at hand would match a sequence of the same length n randomly picked from the genome. In a simple model, this probability can be estimated as $\prod_{j=1}^{n} q_{a_j}$, where a_j is the nucleotide at position j of the sequence at hand and q_a is the probability that the nucleotide a occurs at a randomly selected position in the genome. For example, when finding motifs in a genome with a 40% G + C content the background probabilities q_a are 0.2, 0.3, 0.3 and 0.2 for a = A, C, G or T, respectively. Defining the scores $s_{a,i}$ as log-odds ratios of the target probabilities $p_{a,i}$ and background probabilities q_a

$$s_{a,i} = \ln \frac{p_{a,i}}{q_a} \qquad (2)$$

yields $e^S = \prod_{i=1}^{n} p_{a_i,i} \Big/ \prod_{i=1}^{n} q_{a_i}$ (S is the score of the sequence at hand, see Formula 1), which has a specific meaning—it is the ratio of the probability that the sequence matches the motif model and the probability that the same sequence will be randomly selected from the background (e.g. the complete genome). Hence, the log-odds scores measure not only how good a match is the sequence at hand to the motif represented by the initial alignment (Table 1), but also how uncommon it is in the genomic background. The term PSSM usually refers to this particular type of scores (Table 4). Note that the scores can be negative—the negative scores correspond to nucleotides that are more likely to be found at any given position in the genome than at that particular position of the motif, whereas nucleotides more likely to be found in the motif than the background will have positive scores.

The PSSM representation assigns a numerical score to any sequence of length n. However, converting the score into a binary 'yes/no' classification requires an additional step—setting some score cutoff S_0 such that higher scoring sequences are considered a match whereas lower scoring sequences are not. In the example from Table 1, the 22-mers scoring higher than S_0 will be considered potential Crp

binding sites. Setting the appropriate score cutoff rigorously requires assessments of statistical significance, which are complex and generally involve empirical estimates and/or some assumptions regarding the distribution of the scores. Even when a program provides assessments of statistical significance and automatically sets up an appropriate score cutoff, I would still recommend experimenting with different values and how they affect the number of matching sequences found. In many cases, one has some expectation how many functionally significant motifs of the given type can realistically exist in the genome and how they are distributed (e.g. genes versus intergenic regions), and such prior knowledge can be used to complement statistical criteria.

While PSSM is a more flexible representation of a sequence motif than a consensus sequence and has a probabilistic justification, it also has limitations. It considers the positions of the motif to be independent of each other, which may not necessarily be the case. PSSM also does not allow taking into account gaps (insertions or deletions) in the alignment. Both these issues are (in part) resolved in hidden Markov models (HMMs). HMMs are commonly used to describe protein domains and motifs [2, 3] but less often for nucleotide motifs in DNA. Due to the complexity of the HMM theory and relevant algorithms, I refer the readers interested in the subject to specialized literature [4–6].

Selected software for supervised motif finding

There are numerous tools suitable for finding sequence motifs and patterns represented by a consensus sequence, often utilizing the regular expression (regex) or similar syntax. Regex is a formal language used by computer scientists to describe patterns in texts and regex search is included in some text editors. Programs for finding motifs in DNA and protein sequences using the regex syntax are part of many bioinformatics software packages, including the programs dreg and preg of the EMBOSS package (http://emboss.sourceforge.net/). An advantage of regex syntax is that fast algorithms have been developed for this type of search. At the same time, the regex syntax provides high flexibility in defining text patterns. Disadvantages include that the syntax is relatively complex and, because it was designed for searching regular texts, it does not have options to consider strand symmetry of DNA (e.g. in searching for palindromes and inverted repeats). The latter is resolved in, for example, PatScan [7] and Pattern Locator [8] programs, which allow searching for a consensus sequence as well as more complex patterns based on sequence symmetry (e.g. palindromes, tandem repeats, etc.) or combinations of both. PatScan defines the search patterns as a collection of blocks separated by variable distances, whereas Pattern Locator uses syntax similar to regex. Neither PatScan nor Pattern Locator are built for speed but in most cases speed is not a critical issue. PatScan is available for download as well as through a web interface from Argonne National Lab (http://www-unix.mcs.anl.gov/compbio/PatScan/HTML/patscan.html), although the web version can take a long time to return results. Pattern Locator is available from our laboratory web site (http://www.cmbl.uga.edu/software.html) both for download (requires a compilation on a UNIX/Linux system) and through a web interface. The web interface also provides tools for analysis of distribution of the motif sequences in the genome (see below).

There are many programs that employ some form of the PSSM representation. Some require separate steps to design the PSSM from aligned motifs and search for matching motifs in the sequence (e.g. the MEME suite of programs, http://meme.sdsc.edu/). Programs for PSSM search are often linked to programs for unsupervised motif finding (see below). Dscan (accessible via web interface at http://bayesweb.wadsworth.org/cgi-bin/dscan.pl) is convenient to use but the online version sets a 1 megabase limit on the length of the analyzed sequence. Motif Locator on our lab web site (http://www.cmbl.uga.edu/software.html) is easy to use and provides the same tools for subsequent analyses of distribution of matching motifs as Pattern Locator, but it lacks tools for assessments of statistical significance.

Most readers are probably familiar with sequence logos as a way to display sequence motifs (Figure 1) [9]. In a sequence logo, the height of the stack of letters corresponds to the level of conservation at that particular position measured in terms of information entropy. WebLogo is a web server to generate sequence logos from alignments (http://weblogo.berkeley.edu/[10]). HMM-Logo (http://www.sanger.ac.uk/Software/analysis/logomat-m/[11]) is an extension of the sequence logo concept to motifs represented in the HMM form (that is, including variable length gaps) but it is intended

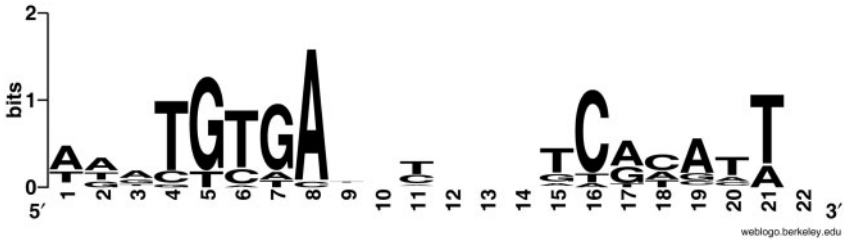

Figure 1: Sequence logo of the Crp binding site. The logo was generated from the alignment in Table 1 by the WebLogo server [10].

for proteins rather than nucleotide sequences. CorreLogo (http://correlogo.abcc.ncifcrf.gov/ [12]) is another variant of the sequence logo concept, specifically designed to display mutual information between different positions in the motif. Like the PSSM representation, the standard sequence logo representation treats all positions of the motif independently and loses all information about mutual relationships between nucleotides at different positions. This information can be very relevant in some cases, particularly for RNA motifs where a nucleotide substitution at one position can be compensated by a complementary substitution in double-stranded segments. CorreLogo displays the mutual information between each pair of positions, and can be particularly useful in analyses of RNA motifs characterized by their secondary structures or any other sequence motifs where interactions among nucleotides at different positions are significant for the motif function.

UNSUPERVISED MOTIF FINDING

Conceptual overview

The unsupervised motif finding techniques are commonly used in discovery of protein binding sites or *cis* regulatory elements. A typical application starts with a collection of genes (or operons) that are presumed to be transcriptionally coregulated. In many cases the coregulated operons share one or more binding sites for transcription factors or other regulatory proteins. Nothing is known about the binding sites themselves or the proteins that bind them. Can one identify the binding sites from the upstream nucleotide sequences alone? Conceptually, the solution is rather simple: find sequence motifs of appropriate length (conserved protein binding sites are often between about 12 and 20 bp long) that are significantly more similar to each other than to other sequences of the same size. These sequences could be the binding sites for a shared regulatory protein. To solve the task, we need (1) a formal measure of mutual similarity among sequence motifs, and (2) an algorithm to find the most similar set of motif sequences with one copy in each upstream nucleotide sequence. The first part can be resolved using the PSSM scores (Formulas (1) and (2)). Consider the situation in Figure 2A with the aim to evaluate the overall similarity of the first motif sequence (marked by the thick bar in the upstream sequence A) to the motif sequences in B, C and D. Using the PSSM representation, we can construct the PSSM from the aligned motif sequences in B, C and D (Formula 2), and calculate the score of the motif sequence in A using this PSSM (Formula 1). The score is a measure of similarity of the first motif sequence to the other three.

The second part of the task consists of finding a combination of motif sequences, one in each upstream sequence, which maximizes the similarity of each motif sequence to the other ones as described above. One could apply a 'brute force' approach and test all possible combinations of motif positions in each of the upstream sequences to find the one that yields the highest score. If the input consists of N upstream sequences all of the same length L and n is the expected motif size, then there are $(L - n + 1)^N$ different combinations to select one motif location in each upstream DNA sequence. Obviously, the number of possible combinations grows rapidly with increasing L and N, making this approach inapplicable in most situations. Hence, the main challenge in solving the task at hand is designing an algorithm that finds the optimal combination quickly without testing all possibilities. The next paragraph briefly describes one solution to this problem—the Gibbs sampling algorithm [13].

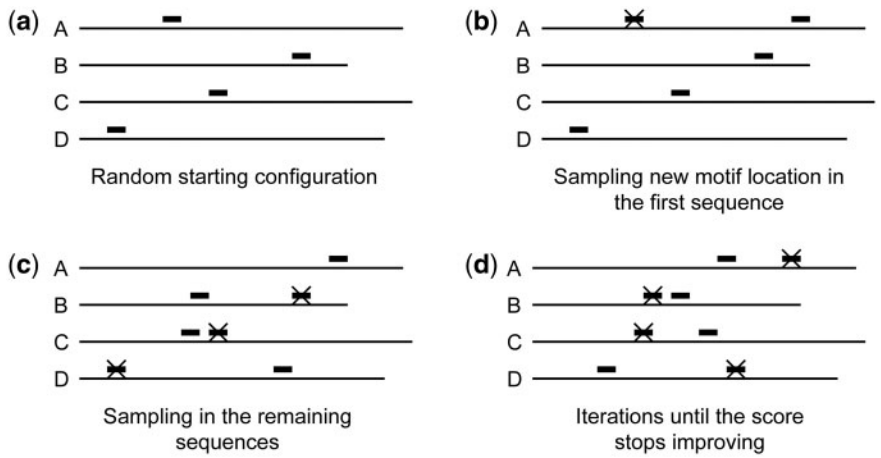

Figure 2: Gibbs sampler algorithm for unsupervised motif finding. (a) Consider searching for a conserved regulatory site in four upstream nucleotide sequences A–D. We assume that each upstream sequence contains one copy of the conserved motif of length 16 bp. The search is started by randomly selecting one 16-mer in each upstream sequence. (b) The 16-mer in the first sequence is dropped from the list and replaced by a new 16-mer. The probability that any particular 16-mer is selected is proportional to its overall similarity to the motif sequences in the other three upstream sequences (see text for details). (c) The process is repeated for sequences B, C and D. (d) This process is iterated until the overall mutual similarity of the motif sequences stops improving.

In addition to the collection of the upstream nucleotide sequences, the Gibbs sampling algorithm requires an estimate of the motif size. We will use 16 bp in this example. Next, we select an initial configuration by randomly picking one 16-mer from each upstream sequence (Figure 2a). At this point the sequences are probably not similar to each other and the PSSM scores will be close to zero. Next we apply the Gibbs sampling strategy to replace the motif sequences one by one by higher scoring ones in an iterative process that gradually improves the mutual similarity of the motif sequences. First, we drop the 16-mer from the upstream sequence A, and use the remaining motif sequences to construct a PSSM (Formula 2). We use this PSSM to assign a score S (Formula 1) to all overlapping 16-mers in the first upstream sequence. Next we select one of these 16-mers to replace the dropped one, with the probability proportional to e^S. That is, the higher the score of the 16-mer the more likely it is that the 16-mer is selected as the new motif location in the upstream sequence A (Figure 2b). Consequently, it is likely that the overall mutual similarity of the motif sequences has slightly improved compared with the initial random selection. This step is repeated with the second upstream sequence and, one by one, with all other upstream sequences, and then iterated until the overall score stops improving (Figure 2C and D). The final configuration is a collection of candidate binding sites of a shared regulatory protein.

What if the initial assumptions about the motif are not satisfied? For example, what if the length of the conserved motif is not exactly 16 nucleotides or some of the upstream sequences do not contain the binding site? Generally, this is not a problem as long as the deviations are not large. The algorithm still does its job, that is, finds the most conserved motif of the given length with one copy in each upstream sequence. If there is no good match in one of the sequences it will simply pick the best match it can find. Most implementations of the Gibbs sampler algorithm include additional steps that allow it to adjust the motif size, filter out poor matches in some of the upstream sequences, or find additional copies of the motif if some of the upstream sequences contain more than one legitimate binding site. Different implementations differ in how they overcome the limitations of the algorithm, and how they define the motif scores and probabilities used in the sampling step. Note that because the algorithm involves a random selection of the starting configuration as well as random sampling during the optimization phase it can return a different result next time it is used with the same input. However, this rarely happens if there is a single conserved motif in the upstream sequence.

Selected software for unsupervised motif finding

My favourite tool for unsupervised motif finding is the Gibbs Motif Sampler (http://bayesweb.wadsworth.org/gibbs/gibbs.html), which offers several implementations of the Gibbs sampling algorithm: site sampler, motif sampler, recursive sampler and centroid sampler. The site sampler is most similar to the simple application of Gibbs sampling strategy described above. It should be used when one expects that a single motif is present in each sequence. Motif sampler is a variant where all input sequences are concatenated and treated as a single long sequence. It is appropriate when the approximate number of copies of the motif in all sequences combined is known, but each sequence can contain different number of copies. The recursive sampler [14] allows more uncertainty in the number of motifs—it should be used when the user expects multiple conserved motifs (e.g. binding sites for different regulatory proteins) with variable number of copies in each sequence. Finally, the centroid sampler [15] employs a more complex sampling strategy that selects not a single optimal motif but a 'centroid solution' which takes into account scores of similar motifs. In practice, if there is a single strong conserved motif in the input sequences all algorithms should be able to find it. On the other hand, weak motifs that are difficult to distinguish from the background noise may be detected only with a single method and optimal parameters. All versions of the program allow imposing additional restrictions under 'advanced options'. For example, many regulatory proteins are homodimers and their binding sites have a palindromic character (Figure 1). In these cases, Gibbs Motif Sampler allows the user to specify parameters that will direct the sampling algorithm towards palindromic motifs.

MEME (http://meme.sdsc.edu/) is another popular tool for unsupervised motif finding. It employs the expectation maximization algorithm [16, 17]. Expectation maximization uses a different sampling strategy but the overall concept is similar to Gibbs sampler—maximize the overall similarity of the motif sequences by gradually improving the scores in an iterative process. MEME can detect several different motifs in the input data, such as binding sites for different transcription factors. In my experience it is less conservative than the Gibbs Motif Sampler (i.e. it reports motifs that Gibbs Motif Sampler does not register), which may or may not be desirable depending on the particular task. MEME is superior to other motif finding programs in elaborate output with convenient links to programs for supervised motif finding, which allow the user to search for additional copies of the motif. Unlike most other programs, the MEME suite supports assessments of statistical significance of combinations of different sequence motifs. For example, it allows searching for a pair of different motifs, such as binding sites for two different regulatory proteins, which occur together.

GLAM [18] and its successors A-GLAM [19] and GLAM-2 (included in the MEME suite) use different optimization algorithms that bypass the need to specify an estimated size of the motif. GLAM-2 can also detect less conserved motifs that include insertions and/or deletions. There are many other programs for unsupervised motif discovery (see, for example, http://zlab.bu.edu/zlab/gene.shtml, http://molbiol-tools.ca/DNA_Motifs.htm or reviews [20, 21]). Tompa and coworkers [22, 23] evaluated performance of several programs (including MEME and GLAM) for their ability to detect eukaryotic transcription factor binding sites using data from the TRANSFAC database [24].

Phylogenetic footprinting

Phylogenetic footprinting is a variant of unsupervised motif finding founded on the expectation that regulatory motifs (e.g. transcription factor binding sites) are subject to selective constraints that make them more conserved than surrounding non-coding sequence [25, 26]. Tools for unsupervised motif finding described above can be applied in phylogenetic footprinting. However, instead of searching for a conserved motif in upstream regions of coregulated genes from the same genome the search is applied to orthologous regulatory regions from different genomes. The general sequence of steps in phylogenetic footprinting is as follows: (1) select a gene for analysis, (2) find orthologous genes in related genomes, (3) identify and extract their regulatory regions and (4) feed them into an appropriate unsupervised motif finding program. There are some caveats. Selecting the right genomes for comparison in step (2) is important for success of this strategy – phylogenetic relatedness, genome size and natural habitat of the organisms used in phylogenetic footprinting affect the accuracy of the results [27]. Generally, selecting closely related genomes works best as

long as the intergenic sequences are not almost identical. Step (3) is complicated by operon structures – the relevant intergenic regions are not necessarily immediately upstream of the orthologous genes but upstream of the whole operons. Since operon boundaries are generally not known one often has to rely on operon predictions in identifying the relevant regulatory regions (for example, http://www.microbesonline.org/operons/[28]). A possible strategy in overcoming the uncertainty in operon boundaries is to include multiple intergenic regions in the search [29], but including the extra sequence decreases the statistical significance of the regulatory motifs. Finally, motif finding algorithms often use a homogeneous statistical model for the background (e.g. the PSSM representation), but as the sequences in phylogenetic footprinting come from different genomes their nucleotide compositions can differ. Due to all these issues, the phylogenetic footprinting strategy does not always yield the desired result but it can identify candidate regulatory motifs in some cases, especially when care is given to finding the optimal sets of orthologous regulatory regions [27].

Software for unsupervised motif finding, including Gibbs Motif Sampler and MEME, can be applied in phylogenetic footprinting. MicroFootPrinter [29] (http://bio.cs.washington.edu/software.html) conveniently automates the selection of orthologous sequences asking the user only to pick a starting gene. However, the MicroFootPrinter does not optimize the selection of genomes for comparison and employs a simple motif finding algorithm that often records an unrealistically large number of candidate motifs.

EXPLORATORY MOTIF FINDING

Conceptual overview

Exploratory motif finding aims to identify sequence motifs that are 'unusual' in statistical terms, some of which may be important for the organism's physiology. The advantage of exploratory motif finding is that it does not require any prior knowledge apart from the sequence itself. The most significant disadvantage is that finding 'unusual' motifs on its own is generally of little use unless combined with additional data and/or experiments to determine the biological function of the motif. Nevertheless, these techniques can serve as hypothesis-generating tools and/or as 'engineering' tools, for example, in identifying suitable genotyping markers (e.g. [30]).

The definition of 'unusual' generally relies on unusual frequency of occurrence—typically unexpectedly high, but it can also be unexpectedly low. I will focus mainly on identification of short dispersed repeats, typically ~8–30 bp length, of high copy numbers, as opposed to general repeat finding tasks, which usually aim to detect long (imperfect) repeats of low copy numbers. Typical algorithms for detecting short repeats of high copy numbers start by counting how many times each possible oligonucleotide of a given length occurs in the sequence and then apply some statistical criteria to assess whether the count is unexpectedly high or low.

Analysis of distribution of sequence motifs in the chromosome can sometimes provide hints regarding their function. Simple questions to ask focus on biased occurrences in genes versus intergenic regions, downstream or upstream of genes, relationship to other relevant markers (for example, a frequent occurrence at a specific distance from starts or ends of genes), or significant associations with particular gene classes (e.g. [31, 32]). Strong periodic patterns in the spacing of sequence motifs with the period close to the DNA helical period (~10.5 bp) can be indicative of motifs involved in DNA interactions with other molecules or motifs that affect the shape of the DNA molecule [32–34].

r-scan statistics is suitable for detecting anomalies in the overall distribution of markers in a DNA (or protein) sequence [35, 36]. One way to detect significant anomalies in the marker distribution is to consider the minimum and maximum distance between a marker and the *r*-th next marker (Figure 3). Formally, if the sequence contains n markers located at positions $x_1, x_2, \ldots, x_n$, then for a given r one can define the values $m^{(r)} = \min(x_{i+r} - x_i)$ and $M^{(r)} = \max(x_{i+r} - x_i)$.

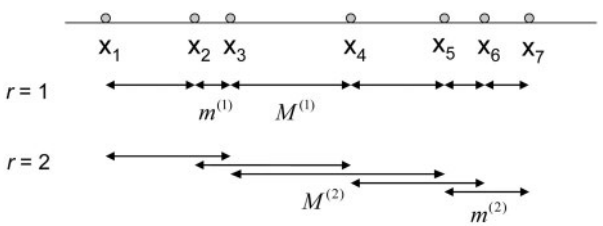

Figure 3: *r*-scan statistics. Finding the minimum and maximum distances between markers for $r = 1$ and 2. The top line represents the nucleotide sequence. Circles signify positions of markers.

Dembo and Karlin [37] derived formulas to estimate the probability that the distances $m^{(r)}$ or $M^{(r)}$ exceed given thresholds if the n markers were distributed randomly, which can be used to determine whether the values $m^{(r)}$ or $M^{(r)}$ are unexpectedly high or low. Too small values of $m^{(r)}$ indicate significant clumping (clustering) of the markers, too large values of $M^{(r)}$ indicate significant overdispersions (or gaps) in the marker distribution, whereas $m^{(r)}$ too large and/or $M^{(r)}$ too small indicate unexpectedly regular distribution. Different values of r can be used to assess the marker distribution at different scales [35, 36].

Distributional anomalies detected by r-scans often have biological interpretations. Figure 4 shows the distribution of DNA uptake signal sequences GCCGTCTGAA/TTCAGACGGC [38] in the *Neisseria gonorrhoeae* NCCP 11945 chromosome. The regions of overdispersion correspond to putative mobile elements including the pathogenicity island GGI (Gonococcal Genetic Island [39]) and putative prophages, and a large cluster of ribosomal protein genes. In another example, an r-scan of DnaA binding sites TTATMCAMA/TKTGKA TAA (M stands for A or C, K for T or G) in the *Escherichia coli* K12 chromosome identifies a statistically significant cluster at the origin of replication, whereas a significant overdispersion is located approximately opposite to the origin (Figure 5).

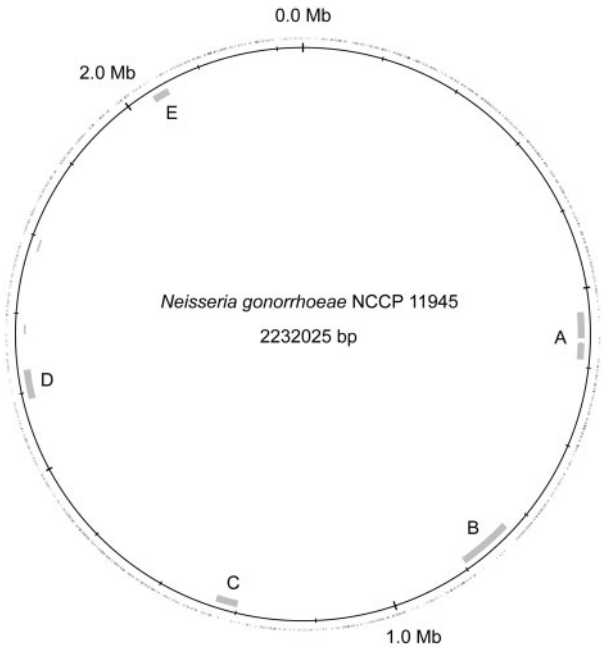

Figure 4: Distribution of DNA uptake signal sequences GCCGTCTGAA/TTCAGACGGC in the *N. gonorrhoeae* NCCP 11945 chromosome. The circle signifies the chromosomal DNA sequence starting clockwise from the top. Marks outside the circle show positions of uptake signal sequences. Grey bars inside the circle signify regions with statistically significant overdispersion of uptake signal sequences; the thicker and thinner bars correspond to 99% and 95% statistical significance, respectively. The region B corresponds to the GGI pathogenicity island whereas regions A, C and D overlap with putative prophages. The region E contains a large cluster of ribosomal protein genes.

Selected software for exploratory motif finding

Significantly overrepresented short motifs can be detected by AIMIE (*Ab Initio* Motif Identification Environment; http://www.cmbl.uga.edu/software/aimie.html [40]), which extends the concept of frequent words proposed by Karlin and coworkers [36, 41]. AIMIE readily detects abundant short motifs such as uptake signal sequences [38], REP

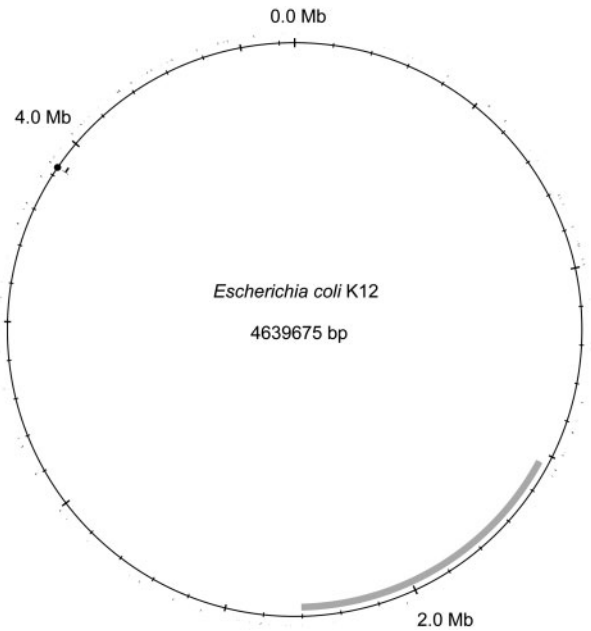

Figure 5: Distribution of DnaA binding sites TTATMCAMA/TKTGKATAA in the *E. coli* K12 chromosome. See legend to Figure 4. Black bars inside the circle mark significant clusters of DnaA binding sites detected by r-scan statistics. The black dot marks the origin of replication located at positions 3923767–3923998.

elements [42], HIP sequences of cyanobacteria [43] or CRISPR [44]. However, it can also pick fragments of longer repeats, such as insertion sequences, if they are sufficiently abundant to qualify as frequent words. AIMIE was developed in our laboratory in an attempt to combine tools for exploratory motif finding and subsequent investigation in a single environment, including *r*-scan statistics and relationship between the sequence motifs and annotated genes. There are many other programs for detection of overrepresented words but most require download and installation on the user's computer. Among those, R'MES (http://migale.jouy.inra.fr/outils/mig/rmes) stands out in providing rigorous methods for statistical assessments of the motif significance [45, 46].

Programs suitable for general repeat finding in the context of complete prokaryotic genomes include REPuter [47], which is accessible via a convenient web interface (http://bibiserv.techfak.uni-bielefeld.de/reputer/submission.html) but can take a long time when applied to a complete bacterial genome (hours to days). Repeat-match is a very fast program for finding exact repeats in complete genomes. It is a modification of MUMmer, which is designed to find matching segments between a pair of different genomes [48]. Repeat-match is available for download with the MUMmer suite of programs (http://mummer.sourceforge.net/). PILER [49] is designed to detect various types of repeats, including tandem repeats, clustered repeats and dispersed repeats such as transposons, but it is geared towards analysis of eukaryotic genomes. Microbiologists might be more interested in PILER-CR [50] and CRISPRFinder [51], which are specifically designed to detect CRISPR repeats in prokaryotes [44]. Both PILER and PILER-CR are available for download (http://www.drive5.com/piler/, http://www.drive5.com/pilercr/), and CRISPRFinder is accessible via web interface (http://crispr.u-psud.fr/Server/CRISPRfinder.php). It is worthwhile to mention Swelfe (http://bioserv.rpbs.jussieu.fr/cgi-bin/swelfe [52]) among general repeat finding programs. Swelfe is a rather unique tool that allows simultaneous detection of repetitive motifs at the level of nucleotide sequence, amino acid sequence and 3-dimensional protein structure (if known). However, this tool is meant for finding internal repeats in individual genes or proteins rather than long DNA sequences such as complete bacterial chromosomes.

CONCLUDING REMARKS AND CAVEATS

Various forms of motif finding are becoming an integral part of a microbiologist's toolbox, together with other computational methods for DNA and protein sequence analysis. When properly used, these techniques can in many cases provide valuable information at a far lower cost than traditional experiments. Described below are some of the caveats related to the use of motif finding algorithms.

In assessing statistical significance, motif-finding algorithms employ various stochastic models to represent the sequence motifs, the genomic sequence background or both. Even the most sophisticated of these models are simplistic when compared with complexities of evolution of native DNA sequences. Moreover, complex stochastic models are not always more appropriate than simple ones because they often involve many parameters that require large amounts of data to estimate accurately. Consequently, in choosing the appropriate method one should consider the task at hand as well as the character of the input data.

Understanding the stochastic models used in a particular method is important for proper interpretation of the results. For example, the *Mycoplasma pneumoniae* genome contains 1218 copies of the hexanucleotide AAAAAA (including overlapping occurrences). This is almost twice as many as ~660 that one would expect in a random sequence generated by the Bernoulli model of independent trials (i.e. reproducing only the overall nucleotide composition of the genome). However, when using a maximum order Markov model (i.e. taking into account the known frequency of the pentanucleotide AAAAA when assessing the expected frequency of the hexanucleotide AAAAAA), the expected number of copies goes up to about 1560. So, is there more AAAAAA than expected or less? The answer depends on the model used, which should be commensurate to the specific question at hand. Ideally, the user should be familiar with the inner workings of the software when using motif finding programs (the same applies to other bioinformatic tools) in order to interpret the results correctly. When this is not practical, a prudent approach is to apply multiple alternative methods and/or parameter settings to evaluate robustness of a particular qualitative result, and interpret with caution the results that are not consistent among different methods.

Key Points

- Supervised motif finding refers to situations when a sample of sequences characterized by a specific biological function is available and the goal is to find other sequences with similar properties.
- Unsupervised motif finding refers to tasks when approximate locations of the motifs are known but not the actual sequences. The most common application relates to finding transcription factor binding sites and other conserved regulatory motifs.
- Phylogenetic footprinting is a variant of unsupervised motif finding, which relies on a higher level of evolutionary conservation of regulatory motifs in comparison to the surrounding intergenic sequences.
- Computational motif finding techniques have limited accuracy and inappropriate applications can lead to misinterpretations of the results and qualitatively wrong conclusions. On the other hand, when used correctly, they can provide helpful information quickly and at a low cost.

Acknowledgements

I thank Drs T. R. Hoover, K. T. Elliott and S. H. Craven for critical reading of the manuscript and constructive comments.

References

1. Altschul SF, Gertz EM, Agarwala R, *et al*. PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic Acids Res* 2009;**37**:815–24.
2. Finn R, Griffiths-Jones S, Bateman A. Identifying protein domains with the Pfam database. *Curr Protoc Bioinformatics* 2003; Chapter 2: Unit 2 5.
3. Mistry J, Finn R. Pfam: a domain-centric method for analyzing proteins and proteomes. *Methods Mol Biol* 2007; **396**:43–58.
4. Eddy SR. Hidden Markov models. *Curr Opin Struct Biol* 1996;**6**:361–5.
5. Eddy SR. What is a hidden Markov model? *Nat Biotechnol* 2004;**22**:1315–6.
6. Schuster-Böckler B, Bateman A. An introduction to hidden Markov models. *Curr Protoc Bioinformatics* 2007;Appendix 3:Appendix 3A.
7. Dsouza M, Larsen N, Overbeek R. Searching for patterns in genomic data. *Trends Genet* 1997;**13**:497–8.
8. Mrázek J, Xie S. Pattern locator: a new tool for finding local sequence patterns in genomic DNA sequences. *Bioinformatics* 2006;**22**:3099–3100.
9. Schneider TD, Stephens RM. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res* 1990;**18**: 6097–100.
10. Crooks GE, Hon G, Chandonia JM, *et al*. WebLogo: a sequence logo generator. *Genome Res* 2004;**14**:1188–90.
11. Schuster-Böckler B, Schultz J, Rahmann S. HMM logos for visualization of protein families. *BMC Bioinformatics* 2004;**5**:7.
12. Bindewald E, Schneider TD, Shapiro BA. CorreLogo: an online server for 3D sequence logos of RNA and DNA alignments. *Nucleic Acids Res* 2006;**34**:W405–11.
13. Lawrence CE, Altschul SF, Boguski MS, *et al*. Detecting subtle sequence signals: a Gibbs.sampling strategy for multiple alignment. *Science* 1993;**262**:208–14.
14. Thompson W, Rouchka EC, Lawrence CE. Gibbs recursive sampler: finding transcription factor binding sites. *Nucleic Acids Res* 2003;**31**:3580–5.
15. Thompson WA, Newberg LA, Conlan S, *et al*. The Gibbs centroid sampler. *Nucleic Acids Res* 2007;**35**: W232–7.
16. Bailey TL, Elkan C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning* 1996;**21**:51–80.
17. Lawrence CE, Reilly AA. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins* 1990;**7**:41–51.
18. Frith MC, Hansen U, Spouge JL, *et al*. Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res* 2004;**32**:189–200.
19. Kim NK, Tharakaraman K, Marino-Ramirez L, *et al*. Finding sequence motifs with Bayesian models incorporating positional information: an application to transcription factor binding sites. *BMC Bioinformatics* 2008; **9**:262.
20. Das MK, Dai HK. A survey of DNA motif finding algorithms. *BMC Bioinformatics* 2007;**8**(Suppl 7):S21.
21. Bulyk ML. Computational prediction of transcription-factor binding site locations. *Genome Biol* 2003;**5**:201.
22. Li N, Tompa M. Analysis of computational approaches for motif discovery. *Algorithms Mol Biol* 2006;**1**:8.
23. Tompa M, Li N, Bailey TL, *et al*. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol* 2005;**23**:137–44.
24. Matys V, Kel-Margoulis OV, Fricke E, *et al*. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* 2006;**34**: D108–10.
25. Duret L, Bucher P. Searching for regulatory elements in human noncoding sequences. *Curr Opin Struct Biol* 1997;**7**: 399–406.
26. Gelfand MS, Novichkov PS, Novichkova ES, *et al*. Comparative analysis of regulatory patterns in bacterial genomes. *Brief Bioinform* 2000;**1**:357–71.
27. McCue LA, Thompson W, Carmack CS, *et al*. Factors influencing the identification of transcription factor binding sites by cross-species comparison. *Genome Res* 2002;**12**: 1523–32.
28. Price MN, Huang KH, Alm EJ, *et al*. A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Res* 2005;**33**:880–92.
29. Neph S, Tompa M. MicroFootPrinter: a tool for phylogenetic footprinting in prokaryotic genomes. *Nucleic Acids Res* 2006;**34**:W366–8.
30. Bruant G, Watt S, Quentin R, *et al*. Typing of nonencapsulated haemophilus strains by repetitive-element sequence-based PCR using intergenic dyad sequences. *J Clin Microbiol* 2003;**41**:3473–80.
31. Guo X, Mrázek J. Long simple sequence repeats in host-adapted pathogens localize near genes encoding antigens, housekeeping genes, and pseudogenes. *J Mol Evol* 2008;**67**: 497–509.

32. Mrázek J. Analysis of distribution indicates diverse functions of simple sequence repeats in *Mycoplasma* genomes. *Mol Biol Evol* 2006;**23**:1370–85.
33. Mrázek J, Gaynon LH, Karlin S. Frequent oligonucleotide motifs in genomes of three streptococci. *Nucleic Acids Res* 2002;**30**:4216–21.
34. Tolstorukov MY, Virnik KM, Adhya S, *et al*. A-tract clusters may facilitate DNA packaging in bacterial nucleoid. *Nucleic Acids Res* 2005;**33**:3907–18.
35. Karlin S, Brendel V. Chance and statistical significance in protein and DNA sequence analysis. *Science* 1992;**257**: 39–49.
36. Karlin S, Mrázek J, Campbell AM. Frequent oligonucleotides and peptides of the *Haemophilus influenzae* genome. *Nucleic Acids Res* 1996;**24**:4263–72.
37. Dembo A, Karlin S. Poisson approximations for *r*-scan processes. *Ann Appl Probab* 1988;**2**:329–57.
38. Smith HO, Gwinn ML, Salzberg SL. DNA uptake signal sequences in naturally transformable bacteria. *Res Microbiol* 1999;**150**:603–16.
39. Dillard JP, Seifert HS. A variable genetic island specific for *Neisseria gonorrhoeae* is involved in providing DNA for natural transformation and is found more often in disseminated infection isolates. *Mol Microbiol* 2001;**41**: 263–77.
40. Mrázek J, Xie S, Guo X, *et al*. AIMIE: a web-based environment for detection and interpretation of significant sequence motifs in prokaryotic genomes. *Bioinformatics* 2008;**24**:1041–8.
41. Cardon LR, Burge C, Schachtel GA, *et al*. Comparative DNA sequence features in two long Escherichia coli contigs. *Nucleic Acids Res* 1993;**21**:3875–84.
42. Higgins CF, McLaren RS, Newbury SF. Repetitive extragenic palindromic sequences, mRNA stability and gene expression: evolution by gene conversion? A review. *Gene* 1988;**72**:3–14.
43. Robinson PJ, Cranenburgh RM, Head IM, *et al*. HIP1 propagates in cyanobacterial DNA via nucleotide substitutions but promotes excision at similar frequencies in *Escherichia coli* and *Synechococcus* PCC 7942. *Mol Microbiol* 1997;**24**:181–189.
44. Sorek R, Kunin V, Hugenholtz P. CRISPR—a widespread system that provides acquired resistance against phages in bacteria and archaea. *Nat Rev Microbiol* 2008;**6**:181–6.
45. Roquain E, Schbath S. Improved compound Poisson approximation for the number of occurrences of any rare word family in a stationary Markov chain. *Adv Appl Probab* 2007;**39**:128–40.
46. Schbath S. An efficient statistic to detect over- and underrepresented words in DNA sequences. *J Comput Biol* 1997;**4**: 189–92.
47. Kurtz S, Choudhuri JV, Ohlebusch E, *et al*. REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res* 2001;**29**:4633–42.
48. Kurtz S, Phillippy A, Delcher AL, *et al*. Versatile and open software for comparing large genomes. *Genome Biol* 2004;**5**:R12.
49. Edgar RC, Myers EW. PILER: identification and classification of genomic repeats. *Bioinformatics* 2005;**21**(Suppl 1): i152–8.
50. Edgar RC. PILER-CR: fast and accurate identification of CRISPR repeats. *BMC Bioinformatics* 2007;**8**:18.
51. Grissa I, Vergnaud G, Pourcel C. CRISPRFinder: a web tool to identify clustered regularly interspaced short palindromic repeats. *Nucleic Acids Res* 2007;**35**:W52–7.
52. Abraham AL, Rocha EP, Pothier J. Swelfe: a detector of internal repeats in sequences and structures. *Bioinformatics* 2008;**24**:1536–7.
53. Robison K, McGuire AM, Church GM. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete Escherichia coli K-12 genome. *J Mol Biol* 1998;**284**:241–54.