Using RSAT oligo-analysis and dyad-analysis tools to discover regulatory signals in nucleic sequences

Matthieu Defrance, Rekin's Janky, Olivier Sand & Jacques van Helden

Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles, Campus Plaine, CP 263, Boulevard du Triomphe, B-1050 Bruxelles, Belgium. Correspondence should be addressed to J.v.H. (jacques.van.helden@ulb.ac.be).

Published online 18 September 2008; doi:10.1038/nprot.2008.98

This protocol explains how to discover functional signals in genomic sequences by detecting over- or under-represented oligonucleotides (words) or spaced pairs thereof (dyads) with the Regulatory Sequence Analysis Tools (http://rsat.ulb.ac.be/rsat/). Two typical applications are presented: (i) predicting transcription factor-binding motifs in promoters of coregulated genes and (ii) discovering phylogenetic footprints in promoters of orthologous genes. The steps of this protocol include purging genomic sequences to discard redundant fragments, discovering over-represented patterns and assembling them to obtain degenerate motifs, scanning sequences and drawing feature maps. The main strength of the method is its statistical ground: the binomial significance provides an efficient control on the rate of false positives. In contrast with optimization-based pattern discovery algorithms, the method supports the detection of under- as well as over-represented motifs. Computation times vary from seconds (gene clusters) to minutes (whole genomes). The execution of the whole protocol should take ~1 h.

INTRODUCTION

This is the second article in a series of four protocols for the analysis of regulatory sequences with the Regulatory Sequence Analysis Tools (RSAT)[1] (http://rsat.ulb.ac.be/rsat/) and biological networks with the Network Analysis Tools (NeAT)[2] (http://rsat.ulb.ac.be/neat/). The first article[3] presents a protocol to predict the location of binding sites for transcription factors (TFs) whose specificity is already known (pattern matching). In the present article, we describe a protocol for the *ab initio* discovery of biological signals in biological sequences (pattern discovery). The third article[4] shows how to write scripts to automate the analysis on multiple clusters of genes using Web services. The fourth[5] describes a workflow for deciphering biological networks by combining network comparison, module identification and path finding.

Several bioinformatics approaches address the problem of motif discovery from a set of input sequences. A typical application is to predict TF-binding motifs by discovering over-represented motifs in promoters of coregulated genes. Many TFs recognize a short oligonucleotide (typically 5–10 bp), with a certain level of accepted substitutions at some positions. Some dimeric TFs recognize dyads, that is, pairs of short oligonucleotide (3–4 bp), separated by a spacing of fixed width but variable content (e.g., CTAn{10}TGG). The discovery of exceptional motifs in biological sequences can play a crucial role in deciphering genome sequences and in interpreting transcriptome data. Several criteria of exceptionality can be considered for selecting relevant motifs: higher/lower frequency than expected by chance (over-/under-representation); concentration at specific positions relative to some reference coordinate (positional bias).

The protocol includes two typical applications of pattern discovery in regulatory sequences: (i) detection of over-represented oligonucleotides and dyads in promoters of coregulated genes (single genome, multiple genes approach) and (ii) discovery of evolutionarily conserved elements in promoters of orthologous genes (single gene, multiple genomes approach).

For consistency, all examples were chosen in bacterial genomes, but the same protocol gives good results with other genomes as well, in particular with fungal genomes. The methods used in this protocol were described in our previous publications[6–9].

Pattern discovery algorithms

Various pattern discovery approaches have been proposed to detect biological signals in nucleotide sequences. Some algorithms rely on probabilistic descriptions of the motifs, namely position-specific scoring matrices (PSSMs), and apply various optimization methods to extract high-scoring motifs. The field was pioneered by Stormo's group, who transposed the concepts from Shannon's information theory to define the theoretical grounds for measuring the conservation of each position of a TF binding[10,11]. The same group developed a greedy algorithm named '*consensus*' for discovering putative TF-binding motifs in unaligned promoter sequences[12,13]. Other machine-learning algorithms have later been adapted to discover PSSM-based motifs in DNA sequences. The program *MEME* is based on an expectation-maximization algorithm[14,15]. The *Gibbs sampling* strategy has been applied to discover motifs in protein sequences[16,17] and later adapted to detect putative TF-binding sites in promoters of coexpressed genes[18–20]. More recent versions of the Gibbs sampling[21,22] support background models based on Markov chains, which take into account the higher order dependencies between adjacent residues in biological sequences.

Another group of algorithms rely on a statistical analysis of oligonucleotide (word) occurrences[6,23–30]. String-based representations of nucleotidic motifs rely either on the 4-letter alphabet (A, C, G, T) or on a 15-letter alphabet (the IUPAC code) that permits to describe partly degenerated positions[30].

The detection of over-represented oligonucleotides is very efficient for detecting various types of functional signals but fails to detect a particular type of TF-binding site—namely the spaced motifs (dyads)—typically recognized by dimeric TFs. Such motifs are particularly important in bacteria because they correspond to the major class of TFs, the helix-turn-helix proteins. In yeast, TFs comprising a fungal zinc cluster domain also recognize spaced

motifs. The program *dyad-analysis* was specifically developed to detect this type of motifs[7]. It performs a systematic analysis of the occurrences for spaced pairs of trinucleotides, with spacing distances varying between 0 and 20. The estimation of dyad over-representation is based on the same binomial statistics as *oligo-analysis*.

Main advantages of oligonucleotide and dyad counting methods

- The statistics on oligonucleotide (word) occurrences are well defined and provide an accurate estimate of the risk of false positives. The significance score associated with predicted motifs provides an indication about their reliability.
- Consequently, programs based on those statistics are able to return a negative answer when one submits some sequences without any specific motifs (negative controls). This is of particular importance for the analysis of high-throughput data because these methods sometimes return very noisy data, and the subsequent analysis (normalization, clustering) may lead to erroneous clusters.
- The search is exhaustive: all possible oligonucleotides or dyads are analyzed, and the method is able to return all the significant motifs in a single run.
- These methods can easily handle large sequence sets. Computing time increases linearly with the sequence length. Whole genomes can be treated within a few minutes.
- The statistical test applies not only to over-represented motifs but also to under-represented motifs, thereby allowing to detect motifs that have been counter-selected in some genome during evolution. In contrast, matrix-based pattern discovery methods are intrinsically unable to detect under-represented motifs.

Main limitations of oligonucleotide and dyads counting methods

- The programs *oligo-analysis* and *dyad-analysis* are restricted to motifs described with the 4-letter nucleotide alphabet (A, C, G, T), plus the N character used for dyad spacing, without explicit treatment of motif degeneracy. As will be shown, motif degeneracy can however be detected because the program can return several oligonucleotides differing by one or a few substitutions. Pattern assembly can then reveal the variable positions in the motifs.
- The motifs are returned in the form of a list of oligonucleotides, rather than as a PSSM[3], which provides a more intuitive description of the position-specific variability of the motif. Lists of oligonucleotides can however be converted into PSSMs in a second step.

Applications of pattern discovery to DNA sequences

In this protocol, we will combine several tools to discover motifs in genome sequences. The flow chart shown in **Figure 1** synthesizes the interconnections between these tools. We will successively present two study cases: (i) detection of over-represented oligonucleotides and dyads in 98 promoters bound by the TF Spo0A of *Bacillus subtilis* and (ii) detection of evolutionarily conserved motifs in promoters of the orthologs of the *purE* gene.

Study case 1. Discovering *cis*-acting elements by detecting over-represented oligonucleotides (option A) or dyads (option B) in promoters of coregulated genes.

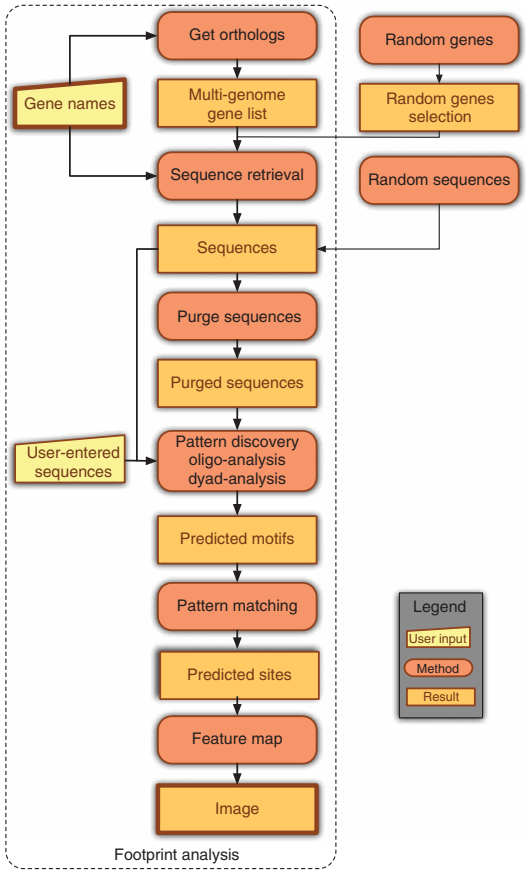


Figure 1 | Flow chart of the tools used in this protocol.

A classical application of pattern discovery is the detection of *cis*-acting elements in promoters of a set of coregulated genes. This application has become very popular since the advent of microarray-based expression profiling[31]. Various methods can be applied to obtain clusters of coexpressed genes from a set of microarray expression profiles[32–34]. The underlying hypothesis is that one or several TFs act in a coordinated fashion on the coexpressed genes. One would thus like to discover, in the promoters of the coexpressed genes, some motif that might reveal binding sites for the putative TFs regulating these genes. One difficulty with this approach is that there is not always a one-to-one correspondence between coexpression and coregulation because a given TF might have indirect effect on a large number of genes via the activation/repression of secondary TFs. Coexpression clusters are typically noisy and can be supposed to contain subsets of genes regulated by different mechanisms.

More direct information on TF-binding regions can be obtained from the more recent ChIP-chip technology, which combines Chromatin Immuno-Precipitation (ChIP) with DNA microarrays (chips). This method has for example been applied to a collection of 102 TFs in the yeast *Saccharomyces cerevisiae* to detect their target genes, first in a rich culture medium[35] and in a second time in various experimental conditions[36]. This ChIP-chip technology provides better evidence about the direct link between a TF and its target genes and permits to further characterize the impact of environmental conditions on the binding specificity.

The application of pattern discovery to promoters of coregulated genes will be illustrated with a first study case. Molle *et al.* combined ChIP-chip technology with microarray expression profiles to detect genes regulated by the TF Spo0A, the main regulator of sporulation in *Bacillus subtilis*[37]. Starting from the list of genes characterized by these experiments, we will show how to use the program *oligo-analysis*[6] to discover the Spo0A-binding motif.

In order to illustrate the benefits of analyzing spaced motifs, we will successively apply *oligo-analysis* (option A) and *dyad-analysis* (option B) to detect over-represented motifs in the promoters of 98 FNR target genes annotated in RegulonDB[38,39].

Study case 2: analyzing evolutionary conservation and divergence between cis-acting elements. The ever-increasing pace of genome sequencing opens a new perspective for the analysis of regulatory sequences: detecting *cis*-acting elements based on their conservation in the promoters of orthologous genes. The first attempt in this direction was proposed by Wasserman and Fickett, who compared large regions of human and mouse genomes, and observed that some conserved fragments in noncoding regions were enriched in TF-binding sites[40]. The conservation of regulatory regions is likely to reflect the constraint to maintain gene regulation during evolution. The underlying model is that mutations have been counter-selected in the sites recognized by TFs, thereby imposing a slower rate of divergence than in surrounding noncoding sequences. Conserved elements found in noncoding fragments are therefore called *phylogenetic footprints*.

The detection of phylogenetic footprints has been applied successfully in bacteria[41–43] and yeast[44]. We recently proposed a method for detecting phylogenetic footprints in bacteria by detecting over-represented dyads in the promoters of orthologous genes. We performed a systematic evaluation of this method and defined its optimal parametric conditions[9]. We also showed that the detection of footprints at various taxonomical levels enables to detect not only conservation but also divergence of *cis*-acting elements during evolution.

This discovery of phylogenetic footprints will be illustrated with our second study case: we will discover motifs in promoters of orthologous genes of the gene *purE* in two distinct bacterial taxa (Enterobacteriales and Bacillales, respectively).

Other applications of this protocol

In this protocol, we selected study cases from bacteria to illustrate the behavior of the programs in well-controlled conditions. In particular, footprint discovery has been well calibrated to return optimal results with bacteria[9]. The pattern discovery tools presented here also give good results with fungal genomes, as shown in our previous publications[6–8,45–48]. They were also used to analyze promoters of other organisms, for example, plants[49] or drosophila[50,51]. In vertebrates, pattern discovery gives much poorer results due to the dispersion of regulatory elements over very large distances and to the heterogeneity of promoter compositions.

Oligonucleotide-counting methods have also been used to detect functional signals in whole genomes. Although the protocol focuses on the discovery of gene-specific TF-binding motifs, we will also show a typical example of genome-wide pattern discovery (ANTICIPATED RESULTS), where we predict restriction sites by detecting under-represented motifs in the whole genomes of *Escherichia coli K12* and *Bacillus subtilis*, respectively.

MATERIALS

EQUIPMENT

- The only requirement to follow this protocol is a personal computer with connection to Internet and a Web browser.

EQUIPMENT SETUP

- Sample data sets used for this protocol can be downloaded from the supporting site on the RSAT Web server http://rsat.ulb.ac.be/rsat/data/published_data/nature_protocols/pattern_discovery/

For the first study case (coregulated genes), two files are provided: upstream sequences of *Bacillus subtilis* genes regulated by *Spo0A* in the file *Bacillus_subtilis_Spo0A_ChIP-chip_target_upseq.fasta* and upstream sequences of 98 FNR target genes of *Escherichia coli K12* in the file *Escherichia_coli_K12_FNR_RegulonDB_target_upseq.fasta*.

PROCEDURE

Application 1: discovering *cis*-acting elements by detecting over-represented oligonucleotides and dyads in promoters of coregulated genes

1| In your Web browser, open a connection to the RSAT Web server (http://rsat.ulb.ac.be/rsat/). Click on the **Pattern discovery** title in the left menu. A sub-menu opens (**Fig. 2**).

2| Choose which type of motifs you would like to detect: oligonucleotides (option A) or dyads (option B).

(A) Discovering *cis*-acting elements by detecting over-represented oligonucleotides in promoters of coregulated genes

(i) Click on the link **oligo-analysis** to open the oligo-analysis form.

(ii) In the sequence section, copy/paste your sequences in the box. To upload sequences from a file, click on the browse button and choose the appropriate file on the computer. For the study case discussed in this protocol, copy and paste *Bacillus subtilis* sequences regulated by *Spo0A* that are provided in the file *Bacillus_subtilis_Spo0A_ChIP-chip_target_upseq.fasta* (see EQUIPMENT SETUP).

(iii) In the section ***Oligomer counting mode***, make sure that that ***Oligomer length*** is set to 6.

(B) Detecting over-represented dyads in promoters of coregulated genes

(i) Click on the link **dyad-analysis** to open the dyad-analysis form.

(ii) In the sequence section, copy/paste or upload your sequences. To illustrate the usefulness of dyad-analysis for detecting spaced motifs, we will analyze the promoters of 98 FNR target genes of *Escherichia coli K12*

(see EQUIPMENT SETUP). Copy the sequences provided in *Escherichia_coli_K12_FNR_RegulonDB_target_upseq.fasta* and paste them in the sequence box.

(iii) In the section ***Dyad counting mode***, make sure that that ***monad length*** is set to 3, ***Spacing*** from 0 to 20 and ***Dyad type*** to ***any dyad***.

! CAUTION The analysis of dyads takes more time than that of single oligonucleotides. For the chosen parameters, the time should be ~20 times longer than for *oligo-analysis*.

3| The ***oligo-analysis*** and ***dyad-analysis*** forms contain a large number of shared options, whose detailed description is available in the online manual. The default parameters are those that usually return interesting results with promoters. We will just check below some of the most important options and explain why they are critical.

4| Make sure that the option ***Purge sequences*** is checked.
▲ CRITICAL STEP Sequence purging is essential to discard redundancy in the sequence set. See **Box 1** for a detailed explanation about sequence redundancy.

5| In the section ***Background model***, the following options should be selected: ***Genome subset***, ***Sequence type upstream-noorf***. Select the ***Organism*** according to your input sequences (depending on the study case, choose *Bacillus subtilis* or *Escherichia coli K12*).
▲ CRITICAL STEP The choice of the background model is one of the most crucial parameters for pattern discovery. An inappropriate background model provokes noisy results that can lead to erroneous interpretations. See **Box 2** for a detailed explanation about background models.
? TROUBLESHOOTING

6| In the section ***Return***, make sure that the box ***Binomial proba*** is checked and that the ***Lower Threshold*** on ***Significance*** is 0.
▲ CRITICAL STEP The programs *oligo-analysis* and *dyad-analysis* can compute various statistics to score the level of over-representation of each oligonucleotide/dyad. The binomial significance is the most appropriate under our working conditions, as explained in **Box 3**.

7| Activate the option ***Convert assembled patterns to Position-Specific Scoring Matrices***.
! CAUTION This option will activate the program *matrix-to-patterns*, which scans the input sequences to build PSSMs from over-represented oligonucleotides. By default, this option is not checked because the scanning phase can be time consuming when large sequences are analyzed (several Mb) or when many motifs are detected. For the analysis of microbial regulons, sequences are generally not too large, and it is thus useful to activate the conversion to PSSMs.

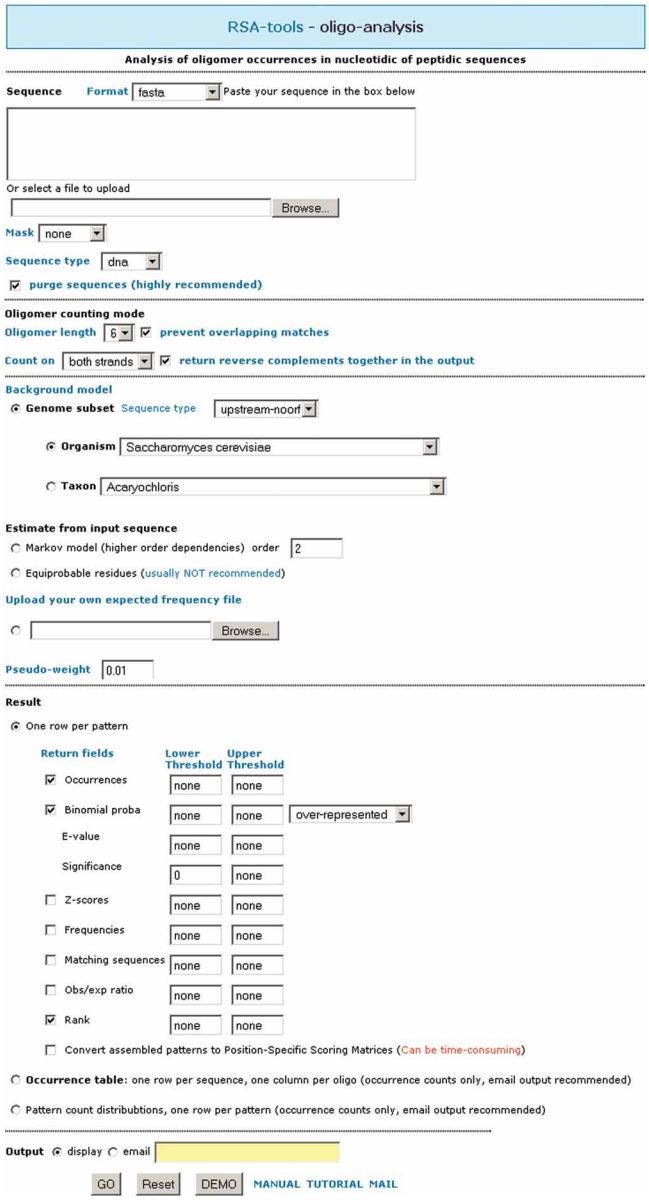


Figure 2 | The Web interface of *oligo-analysis* is divided in five sections separated by dashed lines. The section 'Sequence' allows the user to paste or upload DNA sequences in various standard formats. The second section specifies the counting mode for oligonucleotides (length, strands, etc.). The third section presents a choice of background models (see **Box 2**). The section 'Result' permits the user to select the output fields, and to specify thresholds on various statistics. The section 'Output' allows the user to display the results directly in the Web browser or to store them on the server and send a notice by email when the analysis is finished.

8| Leave other parameters unchanged and click ***GO***.

9| After a few moments, the top of the result page appears. The primary result is a list of over-represented hexanucleotides or dyads, each characterized by various attributes that contributed to estimate its significance (See **Box 3**). Below this list, the section ***Pattern assembly*** indicates that several patterns (*oligonucleotides* or *dyads*) can be assembled to form a larger motif (**Fig. 3b**).
? TROUBLESHOOTING

BOX 1 | PURGING SEQUENCES

Redundant sequences are harmful to pattern discovery because they violate the statistical hypothesis of independence that is essential to the binomial statistics (see **Box 3**).

Sources of sequence redundancy

Redundancy in a sequence set can originate from different sources:

1. **Repetitive elements**. This is especially critical in vertebrates, where a considerable fraction of the genome is made of repetitive elements.
2. **Recent duplications**. The duplication of a genome segment generates two identical sequences. Recently duplicated genes generally ensure the same function and are regulated in the same way, until one of the two copies diverges. It is thus logical to find pairs of recently duplicated genes in sets of coregulated genes.
3. **Promoters of divergently transcribed neighbor genes**. Neighbor genes transcribed in divergent directions share the same promoter sequence. In some cases (but not always), both are regulated by *cis*-acting elements located in their intergenic region. It is thus common to find pairs of neighbor genes in sets of coregulated genes.
4. **Promoters of orthologous genes from very close species**. When analyzing promoters of orthologous genes, the input data may contain several almost identical sequences, obtained from closely related organisms (e.g., ten different strains of the species *Escherichia coli*). This will create a strong redundancy in the input set, which may completely bias the statistical analysis if it is not treated in an appropriate way.

Treatment of sequence redundancy

In order to avoid the problems of redundancy, we recommend masking redundant segments in the input sequences before counting oligonucleotides or dyads. In the Regulatory Sequence Analysis Tools suite, the masking of redundant elements is ensured by the program REPuter[57,58], which replaces repetitive segments by N. By default, we mask all segments of 30 bp that are identical to some other segment of the input data set. Note that the problem of redundancy only concerns the statistical test used for pattern discovery. For pattern matching and feature-map drawing, we generally want to detect all the instances of the patterns. By default, the server masks the repeats for pattern discovery and uses the unmasked sequences for pattern matching.

10| A few moments later, the lower part of the result page displays two types of PSSMs. ***Significance matrices*** directly reflect the assembly of over-represented oligonucleotides, whereas ***Count matrices*** are obtained in a second phase, by scanning the input sequences with the significance matrices.

11| At the bottom of the result page, in the ***Next step*** box, click on the button ***string-based pattern matching (dna-pattern)***. The dna-pattern form is displayed, where all parameters have been automatically filled in to detect the occurrences of the patterns discovered in the previous step. Leave all parameters unchanged and click ***GO***.

12| The program *dna-pattern* returns a list of features indicating the positions of the oligonucleotides/dyads and the limits of the input sequences. We do not actually want to analyze these instances as they are presented in the table but rather to visualize them graphically. At the bottom of the *dna-pattern* result page, click on the ***feature map*** button to open the feature-map form.

13| The ***feature map*** form presents a variety of display options. A good approach is to generate a first figure with the default options, and then to come back to the form in order to refine the display according to your needs. Leave all options unchanged and click ***GO***. The interpretation of the oligonucleotides, their assembly and the feature map will be discussed below (see ANTICIPATED RESULTS).

Application 2: analyzing evolutionary conservation and divergence between *cis*-acting elements

14| In the previous steps, we analyzed the promoters of coregulated genes in a single genome. We will now take the orthogonal approach: starting from a single gene, we will detect motifs in the promoters of orthologous genes. For this, we will use the program *footprint-discovery*, which is actually an automatic workflow combining several RSAT tools (**Fig. 1**). Open a new window in your Web browser, and open a connection to the RSAT Web site (http://rsat.ulb.ac.be/rsat/). In the left panel, expand the ***Pattern discovery*** triangle and click on the link ***footprint-discovery***.

15| At the top of the footprint-discovery form, select the ***Organism***. This is the organism to which your query gene(s) belong (for the study case, select *Bacillus subtilis*).

16| In the box ***Query genes***, enter one or several gene names (for the study case, type purE). If you want to enter multiple gene names, each one should come as the first word of a new line.

! CAUTION When several genes are entered, the program collects promoters of orthologs for all the query genes and submits them altogether to the pattern discovery tools. If you want to analyze several genes separately, you need to run several separate queries.

BOX 2 | BACKGROUND MODEL

The choice of a background model is one of the most crucial parameters for pattern discovery. The background model is used to estimate the oligonucleotide or dyad frequencies that would be expected in a sequence devoid of any specific biological signal (a 'neutral' sequence). The background model can be estimated either from a reference sequence set (background sequences) or from the input sequence itself.

The background model can be estimated from some reference set when the input sequences (the query) are a subset of a larger collection (the reference)[6]. For example, for sets of coregulated genes, expected frequencies are estimated by taking the oligonucleotide frequencies measured in the whole collection of promoters of the selected organism. The Regulatory Sequence Analysis Tools contain predefined background models for oligonucleotides and dyads, with different counting modes (single or both strands, with or without overlapping occurrences), and for each supported organism (currently, >600 species). The Web site also supports taxon-wide background models, which are built by counting oligonucleotide or dyad frequencies in all promoters of all genes of all organisms belonging to a given taxon. Taxon-wide background models are used for the analysis of phylogenetic footprints (i.e., elements conserved in the noncoding regions surrounding a set of orthologous genes).

In some cases, the analysis is performed on sequences for which the concept of external reference set does not apply. This is, for example, the case when analyzing a whole genome or the set of all 3'-untranslated regions for a given organism[8]. In such case, expected frequencies can be estimated from the input sequences themselves (the frequency of a given oligonucleotide or dyad is estimated on the basis of frequencies of the shorter oligonucleotides it is composed of). The simplest type of input-based estimation is a **Bernoulli model**, where the expected frequency of an oligonucleotide is the product of residue frequencies measured in the input sequence set. For example, if the input sequences contain 33% A, 30% T, 18% G and 19% C, the expected frequency of GATCGG is

$$P(\mathrm{GATCGG}) = P(\mathrm{A}) \cdot P(\mathrm{T}) \cdot P(\mathrm{G})^3 \cdot P(\mathrm{C}) = 0.33 \cdot 0.30 \cdot 0.18^3 \cdot 0.19 = 0.0001097,$$

whereas the expected frequency of TATAAA is

$$P(\mathrm{TATAAA}) = P(\mathrm{A})^4 \cdot P(\mathrm{T})^2 = 0.33^4 \cdot 0.30^2 = 0.001067.$$

Note that there is an order of magnitude between the expected frequencies of these two particular oligonucleotides. This shows how important it is to use a relevant method to estimate background probabilities.

Bernoulli models are simple to estimate but rely on an assumption of independence between successive nucleotides, which generally does not hold for biological sequences. For example, it is well known that yeast noncoding sequences contain a higher frequency of poly-AT oligonucleotides than expected from a Bernoulli model. Another well-documented case of dependency between successive residues is the avoidance of CpG dinucleotides in vertebrate sequences (this avoidance is released in CpG islands). Dependencies between successive residues can be represented using **Markov models**, where the expected frequency of an oligonucleotide is estimated on the basis of its composition in shorter oligonucleotides[8]. For a detailed description of Markov models for biological sequence, see ref. 59. Markov models are appropriate for the analysis of whole genomes (**Box 4**) because, in this case, there is no sequence set that could be considered as 'reference' for the random expectation. The order of the Markov model is a delicate choice, which depends on the size of motifs (oligonucleotides) and of the input sequence set. Higher-order models show a better fit to the data, but they require larger training sets to avoid overfitting.

For the analysis of spaced motifs, the program *dyad-analysis* supports an alternative model, called 'monad', where the expected frequency of a dyad is estimated by taking the product of frequencies of its monads (trinucleotides) in the input sequences[7]. For example, $P(\mathrm{CTAn}_{10}\mathrm{TAG}) = F(\mathrm{CTA}) \cdot F(\mathrm{TAG})$.

BOX 3 | SCORING STATISTICS

Various statistics have been proposed to compare observed and expected frequencies (reviewed in ref. 60): observed/expected ratio[61], *z*-score[62], binomial[6], Poisson and compound Poisson[63].

The programs used in this protocol (*oligo-analysis* and *dyad-analysis*) rely on the binomial distribution[6]. For a given oligonucleotide of size k, a sequence of length L is considered a succession of $N = L - k + 1$ trials, corresponding to each position where a k-mer can be found. Each trial results in either a success (an occurrence of the considered oligonucleotide starts at that position) or a failure (no occurrence). The probability to observe at least s successes by chance is given by the inverse cumulative binomial distribution.

$$P(X = s) = C_N^s p^s (1-p)^{N-s} = \frac{N!}{s!(\mathrm{N}-\mathrm{s})!} p^s (1-p)^{N-s}$$

$$P(X \geq s) = \sum_{i=s}^{N} P(X = i)$$

This probability, also called **nominal *P*-value**, indicates the risk for a given oligonucleotide to be considered as significant when it is not. In other terms, this is the risk of false positive for one given oligonucleotide.

Because the same statistical test is applied to all possible oligonucleotides, we perform a correction for multitesting by multiplying the nominal *P*-value by the number of oligonucleotides tested to obtain an ***E*-value** (expected value). The *E*-value represents the number of false positives expected by chance at a given level of *P*-value. The significance score is a minus log transformation of this *E*-value.

$$E_{\mathrm{val}} = D \cdot P_{\mathrm{val}}$$

$$sig = -\log_{10}(E_{\mathrm{val}})$$

a **Result**

```
column headers
        1       seq             oligomer sequence
        2       identifier      oligomer identifier
        3       exp_freq        expected relative frequency
        4       occ             observed occurrences
        5       exp_occ         expected occurrences
        6       occ_P           occurrence probability (binomial)
        7       occ_E           E-value for occurrences (binomial)
        8       occ_sig         occurrence significance (binomial)
        9       rank            rank
        10      ovl_occ         number of overlapping occurrences (discarded from the count)
        11      forbocc         forbidden positions (to avoid self-overlap)
```

seq	identifier	exp_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	forbocc
gtcgaa	gtcgaa|ttcgac	0.0003247830970	20	2.89	4.3e-11	8.9e-08	7.05	1	0	98
tcgaca	tcgaca|tgtcga	0.0003267341665	17	2.91	1.4e-08	2.8e-05	4.55	2	0	85
cgacaa	cgacaa|ttgtcg	0.0003696576964	16	3.29	4.1e-07	8.5e-04	3.07	3	0	80
attcga	attcga|tcgaat	0.0004418472694	13	3.93	0.00023	4.8e-01	0.32	4	0	65

```
 Job started 2008_05_25.014258
; Job done    2008_05_25.014259
```

Pattern assembly

```
; pattern-assembly  -v 1 -subst 1 -top 50 -2str -i public_html/tmp/oligo-analysis.2008_05_25.014252.res -o public_html/
; Input file    public_html/tmp/oligo-analysis.2008_05_25.014252.res
; Output file   public_html/tmp/oligo-analysis.2008_05_25.014252.asmb
; Input score column            8
; Output score column           0
; two strand assembly
; max flanking bases            1
; max substitutions             1
; max assembly size             50
; top number of patterns        100
; number of input patterns      4
;

;assembly # 1   seed: gtcgaa    4 words length
;  alignt          rev_cpl      score
ttgtcg...       ...cgacaa       3.07
.tgtcga..       ..tcgaca.       4.55
..gtcgaa.       .ttcgac..       7.05
...tcgaat       attcga...       0.32
ttgtcgaat       attcgacaa       7.05    best consensus
;Job started 25/05/08 01:43:01 CEST
;Job done    25/05/08 01:43:01 CEST
```

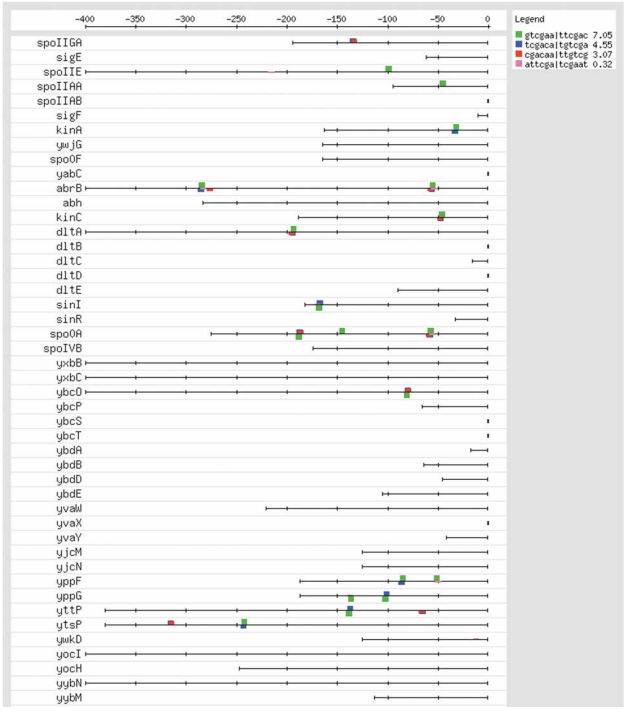
b **Significance matrices**

```
; convert-matrix  -v 1 -i public_html/tmp/oligo-analysis.2008_05_25.015653.asmb -from assembly -return counts,
; Input files
;       input   public_html/tmp/oligo-analysis.2008_05_25.015653.asmb
; Output files
;       output  public_html/tmp/oligo-analysis.2008_05_25.015653_pssm_sig_matrices.txt
; pseudo-weight         1
;
; MATRIX 1/1 : oligo-analysis.2008_05_25.015653
;
;
; Matrix type: counts
; Pos   |       1       2       3       4       5       6       7       8       9
; ----- |-------|-------|-------|-------|-------|-------|-------|-------|-------|
a       |       0       0       0       0       0       0       7.05    7.05    0
c       |       0       0       0       0       7.05    0       0       0       0
g       |       0       0       7.05    0       0       7.05    0       0       0
t       |       3.07    4.55    0       7.05    0       0       0       0       0.32
```

c **Count matrices**

```
; convert-matrix  -v 1 -i public_html/tmp/oligo-analysis.2008_05_25.015653_pssm_sig_sites.ft -from feature
; Input files
;       input   public_html/tmp/oligo-analysis.2008_05_25.015653_pssm_sig_sites.ft
; Output files
;       output  public_html/tmp/oligo-analysis.2008_05_25.015653_pssm_count_matrices.txt
; pseudo-weight         1
;
; MATRIX 1/1 : oligo-analysis
;
;
; Matrix type: counts
; Pos   |       1       2       3       4       5       6       7       8       9
; ----- |-------|-------|-------|-------|-------|-------|-------|-------|-------|
a       |       5       1       1       0       0       3       23      25      9
c       |       1       0       0       0       24      0       0       0       0
g       |       3       0       24      0       0       22      1       0       2
t       |       16      24      0       25      1       0       1       0       14
```

d



17| Select the ***Taxon*** in which orthologs will be collected. For this study case, select Bacillales.
▲ **CRITICAL STEP** The choice of the taxonomical level is one of the most important parameters for the detection of phylogenetic footprints. A too-narrow taxonomical level (e.g., a genus) will contain almost identical sequences, so that the signal will not emerge from the general conservation. If the taxonomical level is too broad (e.g., a whole kingdom), it will encompass genes whose regulation has diverged, so that all the regulatory signals will not be conserved enough to be detected. On the basis of our evaluation[9], we recommend to start the analysis at the level of the order (e.g., Enterobacteriales, Bacillales) or the class (e.g., Gammaproteobacteria).

18| Make sure that the option ***predict operon leader genes*** is not activated.
▲ **CRITICAL STEP** Bacterial genes are organized in operons (a single transcription unit can include multiple coding sequences). Thus, the signals involved in the regulation of a given gene are not always in the sequence immediately upstream of this gene. With the option ***predict operon leader genes***, the program will infer operons and collect promoters upstream of the operon leader genes instead of those found immediately upstream of each orthologous gene. Operon prediction however involves one additional predictive step, which includes a certain rate of errors. A pragmatic approach is to perform a first analysis

Figure 3 | Example of pattern discovery in promoters bound by the transcription factor Spo0A from *Bacillus subtilis*. (**a**) The header of the *oligo-analysis* indicates the parameters used for the analysis. The oligonucleotide table shows the significant oligonucleotides detected in the set of promoters, and their assembly (below the table) indicates that these oligonucleotides reveal different fragments of a larger pattern **`ttGTCGAAt`**. (**b**) Significance matrices obtained from the assembled oligonucleotides. (**c**) Count matrices obtained by scanning input sequences with the significance matrices. (**d**) Feature map of the significant oligonucleotides. Note that the precise values of the oligo-analysis results can slightly vary with successive versions of the genome stored at NCBI (see **Supplementary Fig. 1** online for the full version of this figure).

without operon inference and, in case most promoters would seem very short (and your orthologs would thus probably be located inside operons), redo the analysis with operon inference.

19| Specify the ***dyad filtering*** option according to your needs.
▲ CRITICAL STEP When the option ***dyad filtering*** is checked, the analysis is restricted to the dyads found with at least one occurrence in the promoter of the query gene of the selected organism. This is very useful if your analysis aims at studying the regulation of this particular organism. On the contrary, if you want to analyze the divergence of the regulatory motifs between different branches of a large taxonomical groups (e.g., different classes of bacteria), you should carefully avoid dyad filtering because it would only return the motifs found in the promoter of the 'seed' organism and thereby prevent you from detecting divergent motifs found in more distant organisms. For the study case, we deactivated dyad filtering to discover motifs for different taxa (Bacillales, Enterobacteriales), irrespective of the organism that was considered as 'seed' for this taxon (*Escherichia coli K12*, *Bacillus subtilis*, respectively).

20| Make sure that the option ***background model*** is set to taxfreq.
▲ CRITICAL STEP Our evaluation[9] showed that the 'taxfreq' model generally gives significantly better results than the 'monad' model.

21| In the ***Return fields*** box, check the threshold values. In standard conditions, the ***Lower threshold*** on ***Significance*** should be set to 0 (to select significant motifs only) and the ***Upper threshold*** on ***Rank*** to 50 (this will restrict the result to the 50 top-ranking dyads). Optionally, you can adapt these threshold parameters to increase or reduce the stringency of the analysis.

22| Fill in the ***email*** box. For this program, the only output format is by email notification because the detection of footprints combines several operations that can sum up to several minutes, depending on the load of the server and the number of species in the taxon.

23| Leave all other parameters unchanged and click ***GO***.

24| After a few minutes, you should receive an email indicating the URL of the result. The result page is a short report about the analysis, with links to separate files corresponding to the different steps of the process (collection of orthologs, promoter sequences, significant dyads, assembled dyads, feature map).

25| To analyze the divergence between regulatory elements, perform all the operations above, starting from Step 17, but with a different organism and taxon (for the study case, try the organism *Escherichia coli K12* with the taxon Enterobacteriales). The results will be discussed in the ANTICIPATED RESULTS.

● TIMING
The timing properties were tested on a Macintosh MacBook Pro equipped with a 2.16 GHz Core Duo and 2 Gb RAM. The typical time requirement for the tasks described in this protocol is indicated in **Table 1**.

TABLE 1 | Computation time for typical tasks presented in this protocol.

Task	Algorithm	Conditions	Sequence size (kb)	Time
Over-represented oligonucleotides in promoters of coregulated genes	Oligo-analysis	*Bacillus subtilis*, 49 SpoOA target genes	10	1 s
Over-represented oligonucleotides in promoters of coregulated genes	Oligo-analysis	*Escherichia coli K12*, 99 FNR target genes	27	1 s
Over-represented dyads in promoters of coregulated genes	Dyad-analysis	*Escherichia coli K12*, 99 FNR target genes	27	29 s
Detection of conserved motifs in promoters of orthologous genes	Footprint-discovery (all steps from orthologs identification to feature-map drawing)	Bacillales, 38 orthologs of *Bacillus subtilis* purE	10	3 min 20 s
Genome-scale detection of restriction sites	Oligo-analysis	*Mycoplasma genitalium* whole genome	580	14 s
Genome-scale detection of restriction sites	Oligo-analysis	*Bacillus subtilis* whole genome	4,214	1 min 38 s

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 2**.

TABLE 2 | Troubleshooting table.

Step	Problem	Possible reason	Solution
5	The organism to which your sequences belong does not appear in the pop-up menu ***Organism***	This organism is not (yet) supported on Regulatory Sequence Analysis Tools	If your input sequence is large enough, you can use it to estimate the background model with a Markov chain Alternatively, you can estimate a background model from a separate set of sequences from your organism (reference set), as explained in the *oligo-analysis* manual
9	The program *oligo-analysis* returns too many patterns (e.g., several dozens of patterns)	The background model was not appropriate	Check that the organism and background model sequence type correspond to your input sequences
	The program *dyad-analysis* does not return any result after a few minutes	This may happen for large sequences because the analysis of all dyads costs more time than the analysis of single oligonucleotides	Use the email output

ANTICIPATED RESULTS

Application 1 option A: detection of over-represented oligonucleotides in promoters of coregulated genes

The *oligo-analysis* result file starts with a header indicating the parameters of the analysis, followed by a list of oligonucleotides sorted by significance (**Fig. 3a**). In the promoters of the Spo0A target genes, four pairs of hexanucleotides were significantly over-represented among the 2,080 possible pairs of reverse complements. The top-ranking oligonucleotide, GTCGAA, is found 20 times in the input set, whereas 2.89 occurrences would be expected by chance. The *P*-value (occ_P $= 4.3 \times 10^{-11}$) indicates the probability to observe at least 20 occurrences when 2.89 are expected. The corresponding expected number of false positives is very low (occ_E $= 8.9 \times 10^{-8}$), indicating that such a level of over-representation is very unlikely to result from chance. In this case, the most obvious explanation is that the over-representation of this motif reflects the fact that it is bound by the Spo0A TF.

The *Pattern assembly* section of the result page (**Fig. 3a**) shows that the four significant hexanucleotides can be assembled to form a single motif: TTGTCGAAT (the second column of the assembly shows the reverse complement ATTCGACAA). This 9-mer corresponds to the experimentally characterized consensus of the *Bacillus subtilis* TF Spo0A[37,52,53]. The top-ranking hexanucleotide (GTCGAA) corresponds to the most conserved part of the annotated motif (ttt**GTCGAA**aaa).

For this study case, we activated the option to convert assembled patterns into PSSMs. This is performed in two steps. First, a *significance matrix* is built from assembled oligonucleotides (**Fig. 3b**). The numbers in this matrix indicate the highest significance obtained for each residue (row) at each position of the assembly (column). Second, the input sequences are scanned with the significance matrices to collect all sites with *P*-value below a given threshold (see accompanying article[3]), and a *count matrix* is built with those sites (**Fig. 3c**). The count matrix better reflects position-specific variability of the motif. For the Spo0A motif, we can see that its center (that corresponds to the most significant hexanucleotide) is highly conserved in the matrix, whereas its flanks are partly degenerated. The count matrix can be used to scan new sequences for putative instances of the discovered motif.

The feature map (**Fig. 3d**, **Supplementary Fig. 1** online) shows the matching positions of the significant patterns. Each box represents one hexanucleotide, with a height proportional to the binomial significance. There is a strikingly high frequency of overlap between boxes, indicating that the significant hexanucleotides reveal overlapping fragments of the same motif. The presence of such clumps of mutually overlapping hexanucleotides is usually a good indication for the relevance of the discovered oligonucleotides, and they generally reveal putative binding sites for the TF of interest. The feature map thus helps to refine the interpretation of the discovered motifs and to suggest candidate sites for further experimental validation.

a

seq	identifier	exp_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	forbocc
aatttg	aatttg|caaatt	0.0010822094884	53	28.54	2.7e-05	5.6e-02	1.25	1	2	265
atcaat	atcaat|attgat	0.0011703765749	56	30.87	3e-05	6.3e-02	1.20	2	2	280
atcaaa	atcaaa|tttgat	0.0010398892870	50	27.43	6.8e-05	1.4e-01	0.85	3	0	250
acaaat	acaaat|atttgt	0.0011368730820	53	29.99	9.1e-05	1.9e-01	0.72	4	0	265
atttga	atttga|tcaaat	0.0008371049881	42	22.08	0.00010	2.1e-01	0.67	5	0	210
caaatc	caaatc|gatttg	0.0006819309161	35	17.99	0.00024	5.0e-01	0.30	6	0	175

```
 Job started 2008_06_27.113840
; Job done    2008_06_27.113842
```

Pattern assembly

```
; pattern-assembly  -v 1 -subst 1 -top 50 -2str -i public_html/tmp/oligo-analysis.2008_06_27.113833.res -o public_html/tmp/oligo-analysis.2
; Input file    public_html/tmp/oligo-analysis.2008_06_27.113833.res
; Output file   public_html/tmp/oligo-analysis.2008_06_27.113833.asmb
; Input score column           8
; Output score column          0
; two strand assembly
; max flanking bases           1
; max substitutions            1
; max assembly size            50
; top number of patterns       100
; number of input patterns     6
;

;assembly # 1   seed: aatttg    4 words length
;alignt rev_cpl score
aatttg. .caaatt 1.25
gatttg. .caaatc 0.30
.atttgt acaaat. 0.72
.atttga tcaaat. 0.67
aatttgt acaaatt 1.25    best consensus

;assembly # 2   seed: atcaat    2 words length 7
;alignt rev_cpl score
atcaat  attgat  1.20
atcaaa  tttgat  0.85
atcaat  attgat  1.20    best consensus
;Job started 27/06/08 11:38:44 CEST
;Job done    27/06/08 11:38:45 CEST
```

b

sequence	identifier	expected_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	all_occ	ov_coef	remark
tgan{6}tca	tgan{6}tca|tgan{6}tca	0.0007406876899	57	18.73	1.9e-12	8.5e-08	7.07	1	0	57	1.0000	inv_rep
tgan{7}caa	tgan{7}caa|ttgn{7}tca	0.0008679483941	57	21.84	5.9e-10	2.6e-05	4.59	2	14	71	1.0000	
gatn{5}tca	gatn{5}tca|tgan{5}atc	0.0009443576965	56	23.99	3.2e-08	1.4e-03	2.85	3	24	80	1.0000	
gatn{6}caa	gatn{6}caa|ttgn{6}atc	0.0007293794809	46	18.44	8.6e-08	3.7e-03	2.43	4	7	53	1.0000	
ttgn{8}caa	ttgn{8}caa|ttgn{8}caa	0.0004150917410	29	10.40	2.5e-06	1.1e-01	0.97	5	0	29	1.0000	inv_rep
gatn{4}atc	gatn{4}atc|gatn{4}atc	0.0004227240113	29	10.81	4.6e-06	2.0e-01	0.70	6	0	29	1.0000	inv_rep
attn{5}tac	attn{5}tac|gtan{5}aat	0.0007480060962	42	19.00	5.6e-06	2.5e-01	0.61	7	0	42	1.0000	
tgtn{5}aaa	tgtn{5}aaa|tttn{5}aca	0.0014380417200	67	36.53	7.1e-06	3.1e-01	0.51	8	0	67	1.0000	
atcn{1}aat	atcn{1}aat|attn{1}gat	0.0005662350255	34	14.71	1.5e-05	6.4e-01	0.19	9	1	35	1.0039	

```
Job started 27/06/08 11:43:22 CEST
;Job done    27/06/08 11:43:53 CEST
```

Pattern assembly

```
; pattern-assembly  -v 1 -subst 0 -top 50 -2str -i public_html/tmp/dyad-analysis.2008_06_27.114314.res -o public_html/tmp/dyad-analysis.200
; Input file    public_html/tmp/dyad-analysis.2008_06_27.114314.res
; Output file   public_html/tmp/dyad-analysis.2008_06_27.114314.asmb
; Input score column           8
; Output score column          0
; two strand assembly
; max flanking bases           1
; max substitutions            0
; max assembly size            50
; top number of patterns       100
; number of input patterns     9
;

;assembly # 1   seed: tgannnnnntca      9 words length
;       alignt          rev_cpl  score
ttgnnnnnnntca.  .tgannnnnnncaa  4.59
ttgnnnnnnatc..  ..gatnnnnnncaa  2.43
ttgnnnnnnnncaa  ttgnnnnnnnncaa  0.97
.tgannnnnntca.  .tgannnnnntca.  7.07
.tgannnnnnncaa  ttgnnnnnntca.  4.59
.tgannnnnatc..  ..gatnnnnntca.  2.85
..gatnnnnntca.  .tgannnnnatc..  2.85
..gatnnnnnncaa  ttgnnnnnnatc..  2.43
..gatnnnnatc..  ..gatnnnnatc..  0.70
ttgatnnnnatcaa  ttgatnnnnatcaa  7.07    best consensus

; Isolated patterns: 3
;alignt rev_cpl score
attnnnnntac     gtannnnnaat      0.61    isol
tgtnnnnnaaa     tttnnnnnaca      0.51    isol
atcnaat attngat 0.19    isol
;Job started 27/06/08 11:43:56 CEST
;Job done    27/06/08 11:43:56 CEST
```

c

Figure 4 | Comparison between *oligo-analysis* and *dyad-analysis* for the FNR regulon. (**a**) Significant oligonucleotides and (**b**) dyads detected in promoters of the FNR regulon. (**c**) FNR-binding motif annotated in RegulonDB.

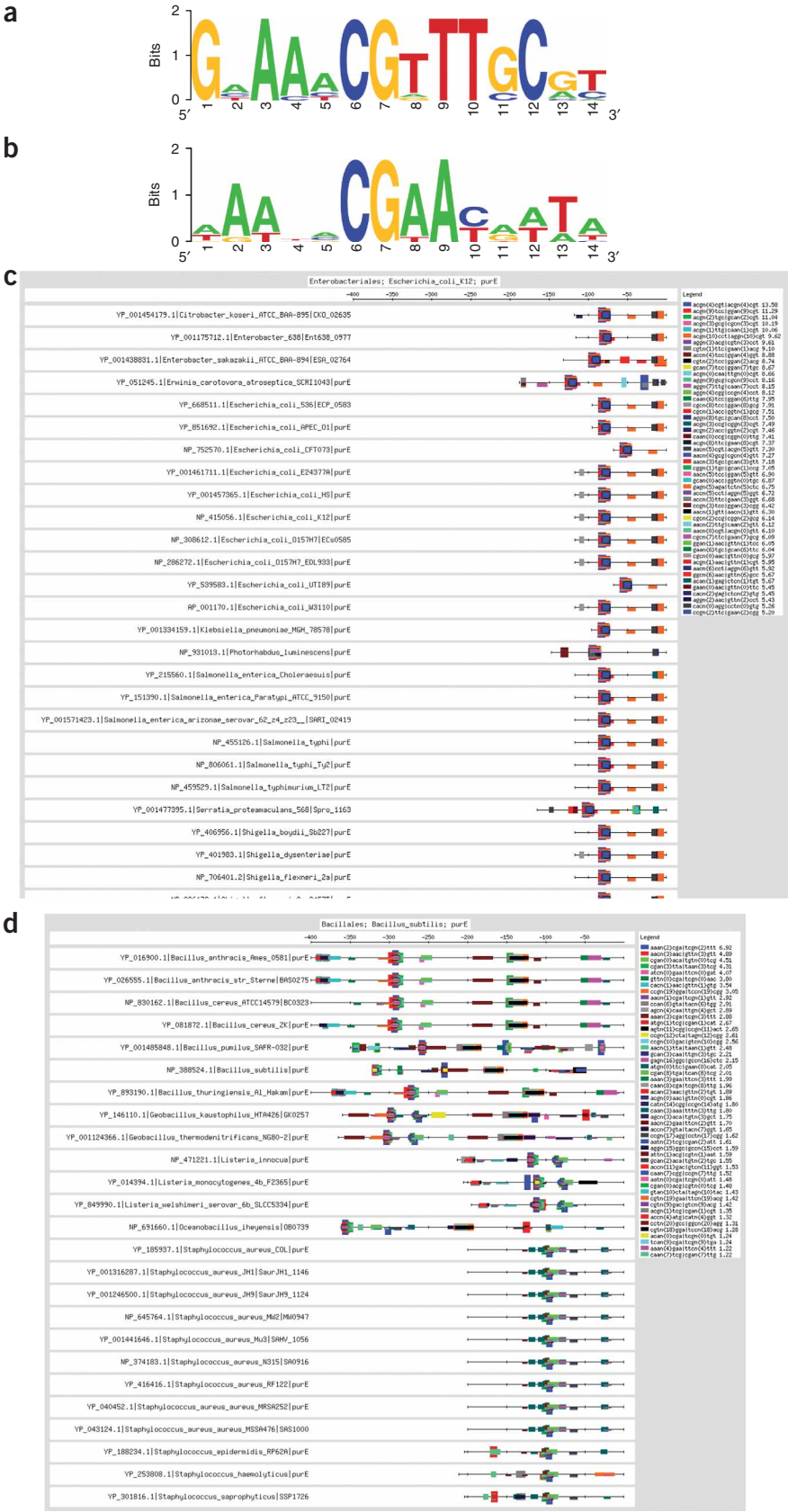


Figure 5 | Example of pattern discovery in promoters of orthologous genes. (**a**,**b**) Sequence logo of binding motifs annotated for the transcription factor PurR in (**a**) *Escherichia coli* and (**b**) *Bacillus subtilis*, respectively. Notice the divergence between the two motifs. (**c**,**d**) Feature maps of the significant dyads detected in *purE* promoters of (**c**) Enterobacteriales and (**d**) Bacillales, respectively (See **Supplementary Fig. 1** online for the full version of **Fig. 5c**).

TABLE 3 | Motifs found by assembling the most significant dyads detected in purE promoters of Enterobacteriales and Bacillales, respectively.

Enterobacteriales

;assembly # 1	seed: acgnnnncgt	11 words	Length
; alignt	rev_cpl	score	
aacnnnnncgt.	.acgnnnnngtt	7.30	
aacnnnngcg..	..cgcnnnngtt	7.27	
aacnnntgc...	...gcannngtt	7.18	
.acgnnnncgt.	.acgnnnncgt.	13.58	
.acgnntgc...	...gcanncgt.	11.04	
.acgnnngcg..	..cgcnnncgt.	10.19	
.acgnnnnngtt	aacnnnnncgt.	7.30	
..cgcnnncgt.	.acgnnngcg..	10.19	
..cgcnnnngtt	Aacnnnngcg..	7.27	
...gcanncgt.	.acgnntgc...	11.04	
...gcannngtt	aacnnntgc...	7.18	
aacgcatgcgtt	**aacgcatgcgtt**	**13.58**	**best consensus**

Bacillales

;assembly # 1	seed: aaanncga	15 words	length
; alignt	rev_cpl	score	
taannntcg.....	.....cgannntta	4.31	
.aaannncga.....	.....tcgnnnttt.	2.88	
.aaannnngaa.....	.....ttcnnnnttt.	1.22	
..aaanncga.....	.....tcgnnttt..	6.92	
..aaannngaa.....	.....ttcnnnttt..	1.99	
...aacnnnaac....	....gttnnngtt...	4.89	
...aacncga.....	.....tcgngtt...	2.92	
...aacnngaa.....	.....ttcnngtt...	1.70	
.....tcgaac....	....gttcga.....	3.80	
.....cgaaca...	...tgttcg.....	4.51	
.....cgannntta	taannntcg.....	4.31	
.....cgancat..	..atgntcg.....	2.67	
.....cgannatt.	.aatnntcg.....	1.61	
.....gaacat..	..atgttc.....	2.05	
.....aacntta	taangtt.....	2.48	
taaaactcgaacatta	**taatgttcgagtttta**	**6.92**	**best consensus**

TABLE 4 | Restriction enzymes and their sites, as annotated in REBASE[55,56].

Escherichia coli restriction enzymes and sites*	
*Cfr*I	YGGCCR
*Eco*105I	TACGTA
*M.Eco*248534P	GTCGAC
*Eco*75KI	GRGCYC
*Eco*168I	GGYRCC
*Eco*101I	GGTCTC
*Eco*149I	GGTACC
*Eci*EI	GGGCCC
*Nar*I	GGCGCC
*Eco*143I	GCGCGC
*Nae*I	GCCGGC
*Eco*17I	GATATC
*Eco*ICRI	GAGCTC
*Eco*159I	GAATTC
*Eco*27kI	CYCGRG
*Eco*161I	CTGCAG
*Eco*E24377ORF3323P	CTGATG
*Eco*112I	CTGAAG
*Eco*52I	CGGCCG
*M.Eco*KO157ORF1953P	CGATCG
*Eco*130I	CCWWGG
*Eco*29kI	CCGCGG
*M.Eco*APECORFBP	CAGCTG
*M.Eco*P15I	CAGCAG
*Eco*72I	CACGTG
*M.Eco*536ORF3355P	ATGCAT
*Eco*255I	AGTACT
*Eco*1524I	AGGCCT
*Eco*47III	AGCGCT
*Eco*VIII	AAGCTT
Bacillus subtilis restriction enzymes and sites**	
*Bsp*MII	TCCGGA
*Hgi*JII	GRGCYC
*Bam*H1	GGATCC
*Pst*I	CTGCAG
*Xho*I	CTCGAG
*Cla*I	ATCGAT

*Hexanucleotidic restriction sites *in Escherichia coli K12*. Note that only one enzyme is shown per site, here, whereas some sites are recognized by several enzymes. **Enzymes from *Bacillus subtilis*.

Another observation is that several sequences are devoid of predicted binding sites. Some of these site-less sequences are very short (spoIIAB, dltB, dltC, etc.) and correspond to intergenic sequences located inside operons. Some larger promoters, however, also lack the Spo0A-binding motif, suggesting that they may either contain variants of that motif or not be directly regulated by Spo0A.

The results obtained with the Spo0A regulon are particularly clean, in that all the significant oligonucleotides correspond to segments of the known motif. The situation is unfortunately not always so clear cut, as will be discussed in the next section.

Application 1 option B: detection of over-represented dyads in promoters of coregulated genes

With the FNR regulon, we tested both options 1A and B of this protocol for the sake of comparison between *oligo-analysis* and *dyad-analysis*. *oligo-analysis* returns seven significant hexanucleotides (**Fig. 4a**), which can be assembled to form a partly degenerated motif: A[T/A]CAA[T/A]T[T/C]. This motif corresponds to the half-site of the FNR-binding motif annotated in RegulonDB (**Fig. 4c**). The most significant hexanucleotide (AATTTG) has a relatively weak significance (sig = 1.25; *E*-value = 0.06). In contrast, the analysis of dyads reveals a highly significant spaced motif (**Fig. 4b**). The most significant dyad, TGAn{6}TCA (sig = 7.07, *E*-value = 8.5×10^{-8}), corresponds to the core of the annotated FNR-binding motif (**Fig. 4c**). This dyad can be assembled with several other ones to form a complete description of the spaced motif bound by FNR:

BOX 4 | DISCOVERING FUNCTIONAL SIGNALS BY GENOME-WIDE DETECTION OF EXCEPTIONAL OLIGONUCLEOTIDES

Oligonucleotide-counting methods can be used to detect exceptional motifs in whole genomes. The term 'exceptional' is taken here in the sense of either over-represented or under-represented (for a didactic presentation on the detection of exceptional motifs, see refs. 59,60). The detection of over-represented oligonucleotides can reveal genome-specific signals such as the CHI motif[23,64]. In contrast, under-represented oligonucleotides can reveal some signals that are avoided in genomes because they would be harmful for the organism. This is, for example, the case of restriction sites, where restriction enzymes specifically bind to cleave DNA[65].

As an example, we show the result of a genome-wide analysis of under-represented hexanucleotides in the whole genomes of the bacteria *Escherichia coli K12* and *Bacillus subtilis*, respectively. The full genome sequences are provided on the supporting Web site, and the analysis can be reproduced by setting the following parameters in the oligo-analysis form:

- **oligonucleotide size** = 6
- **Background model** = **Markov**, **order** 4
- **pseudo-weight** = 0
- **Binomial proba** -> **under-represented**
- **Lower threshold** on **Significance** = **none**
- **Output** -> **email**

The most significant predicted motifs can then be compared with the sites recognized by the restriction enzymes of these organisms.

For the analysis of the whole genome of *Escherichia coli K12*, the result file contains all the possible hexanucleotides grouped by pairs of reverse complement. The top-ranking hexanucleotides (i.e., the most significantly under-represented) are displayed in **Figure 6a**. The most significantly under-represented hexanucleotide is GGCGCC, which corresponds to the binding site of the restriction enzyme *Nar*I (**Table 4**, *Escherichia coli* restriction enzymes and sites). The whole genome of *Escherichia coli* contains no more than 82 occurrences of this hexamer, whereas 1,965 occurrences would be expected by chance, according to the selected background model (Markov model of order 4). The *P*-value (probability to observe such a low number of occurrences by chance) is below the computation limits of the program (P-value $< 1 \times 10^{-300}$) and is thus rounded to 0. Similarly, the site GCCGGC, found in 258 occurrences, whereas 1,609 are expected, corresponds to the restriction site for *Nae*I. The next most significantly under-represented word, AAAAAA, is not related to restriction enzymes. This word is quite frequent (4,896) but much less than expected by chance (7,382 occurrences). Most of the next significant hexanucleotides are palindromes and correspond to well-characterized restrictions sites of *E. coli* (**Table 4**, *Escherichia coli* restriction enzymes and sites), as annotated in the REBASE database[55,56].

For *B. subtilis* (**Fig. 6b**), the most significantly under-represented hexanucleotides contain some AT-rich hexanucleotides (AAAAAA, CTTTTC), together with the sites recognized by the restriction enzymes (**Table 4**, *Bacillus subtilis* restriction enzymes and sites), such as *Bam*HI (GGATCC, sig = 61.11), *Pst*I (CTGCAG, sig = 12.50) and *Cla*I (ATCGAT, sig = 4.13). Some other *B. subtilis* restriction sites are only weakly under-represented, for example *Xho*I (CTGCAG, sig = 1.29). The *Bsp*MII site (TCCGGA) is found in 1,189 occurrences and expected in 1,266 occurrences (not shown on **Fig. 6b**). Despite being less frequent than expected, this site cannot be considered significant because this level of under-representation corresponds to a negative significance (sig = −1.77), indicating an *E*-value superior to 1 ($E_{val} = 59$).

In summary, the genome-wide detection of under-represented hexanucleotides allows the user to detect highly significant genome-specific signals, some of which correspond to well-characterized restriction enzymes of the two model organisms analyzed here (*E. coli* and *B. subtilis*). This approach thus provides a powerful way to investigate newly sequenced bacterial genomes to discover organism-specific restriction sites and other biological signals.

TTGATn{4}ATCAA (see the pattern assembly result on **Fig. 4b**). This example illustrates the interest of analyzing spaced motifs, which are especially important for bacterial and fungal regulation.

Application 2: evolutionary conservation and divergence between *cis*-acting elements

Table 3 displays the most significant motifs discovered with *dyad-analysis* in *purE* promoters of Enterobacteriales and Bacillales, respectively. The most significant dyads detected in Enterobacteriales can be assembled to form the motif **AACGcaTGCGTt** (**Table 3**, Enterobacteriales), which shows a good correspondence with the consensus of the PurR TF annotated for *Escherichia coli* in RegulonDB[38,39]: **gaaAACGttTGCGT** (**Fig. 5a**). The two differing residues in the middle of the motif might reflect differences between intra-species variability (PurR-binding sites in different genes of *E. coli*) and inter-species variability (evolutionary changes within the binding sites of a single gene, *purE*). The dyads detected in promoters of Bacillales form the pattern **taAAActCGAACATTa** (**Table 3**, Bacillales), corresponding to the binding motif of the *Bacillus subtilis* PurR protein annotated in DBTBS[54]: **AAAnnCGAA[CT][AG][AT][TA][AT]** (**Fig. 5b**). The feature maps further reveal that the instances of the significant dyads occupy conserved positions in the *purE* promoters of Enterobacteriales (**Fig. 5c** and **Supplementary Fig. 2** online), whereas in Bacillales (**Fig. 5d**), binding site positions differ between the genera Bacillus and Staphylococcus. The detection of phylogenetic footprints thus reveals the conservation of the binding sites within each taxon and their divergence between two very distant taxa.

a

column headers

1	seq	oligomer sequence
2	identifier	oligomer identifier
3	exp_freq	expected relative frequency
4	occ	observed occurrences
5	exp_occ	expected occurrences
6	occ_P	occurrence probability (binomial)
7	occ_E	E-value for occurrences (binomial)
8	occ_sig	occurrence significance (binomial)
9	rank	rank
10	ovl_occ	number of overlapping occurrences (discarded from the count)
11	forbocc	forbidden positions (to avoid self-overlap)

seq	identifier	exp_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	forbocc
gccggc	gccggc|gccggc	0.0003573055893	258	1609.16	0	0e+00	999.00	1	0	2580
ggcgcc	ggcgcc|ggcgcc	0.0004256374408	82	1916.90	0	0e+00	999.00	2	0	820
aaaaaa	aaaaaa|tttttt	0.0016390737298	4896	7381.71	3e-209	1e-205	204.96	3	1396	24480
agcgct	agcgct|agcgct	0.0003745187858	752	1686.68	2e-144	8e-141	140.12	4	0	7520
ctgcag	ctgcag|ctgcag	0.0004276601823	925	1926.01	2e-142	7e-139	138.14	5	0	9250
cggccg	cggccg|cggccg	0.0001752831094	250	789.40	1e-111	6e-108	107.25	6	0	2500
ccgcgg	ccgcgg|ccgcgg	0.0002911010722	634	1311.00	3e-96	1.2e-92	91.91	7	0	6340
gagacc	gagacc|ggtctc	0.0001559202114	235	702.20	2.5e-93	1.0e-89	88.99	8	0	1175
tccgga	tccgga|tccgga	0.0003544975387	854	1596.51	1e-92	4.2e-89	88.37	9	0	8540
cacgtg	cacgtg|cacgtg	0.0001073988071	122	483.68	3.8e-86	1.6e-82	81.81	10	0	1220
gagccc	gagccc|gggctc	0.0001672589397	295	753.27	4.5e-81	1.8e-77	76.74	11	0	1475
gggccc	gggccc|gggccc	0.0000750329100	61	337.92	7.6e-77	3.1e-73	72.51	12	0	610
cccggg	cccggg|cccggg	0.0001905898589	396	858.34	6.5e-70	2.6e-66	65.58	13	0	3960
gagctc	gagctc|gagctc	0.0001011315302	139	455.45	7.8e-68	3.2e-64	63.50	14	0	1390
gtcgac	gtcgac|gtcgac	0.0002276085625	522	1025.06	1.1e-67	4.6e-64	63.34	15	0	5220
gcatgc	gcatgc|gcatgc	0.0002398058745	563	1079.99	1.9e-67	7.9e-64	63.10	16	3	5630
tggcca	tggcca|tggcca	0.0002501093155	598	1126.39	3.1e-67	1.3e-63	62.90	17	0	5980
gcgcgc	gcgcgc|gcgcgc	0.0007046383334	2269	3173.40	1.7e-64	6.8e-61	60.17	18	191	22690
cttttc	cttttc|gaaaag	0.0007143147258	2375	3216.98	6e-55	2.5e-51	50.61	19	5	11875
attata	attata|tataat	0.0003292469412	957	1482.79	1.5e-48	6.3e-45	44.20	20	46	4785
cagctg	cagctg|cagctg	0.0005293041322	1720	2383.77	1.1e-46	4.4e-43	42.36	21	0	17200
ctgaag	ctgaag|cttcag	0.0008012245683	2792	3608.39	9.1e-46	3.7e-42	41.43	22	0	13960
aggcct	aggcct|aggcct	0.0001808089374	446	814.29	1.9e-45	7.6e-42	41.12	23	0	4460

b

column headers

1	seq	oligomer sequence
2	identifier	oligomer identifier
3	exp_freq	expected relative frequency
4	occ	observed occurrences
5	exp_occ	expected occurrences
6	occ_P	occurrence probability (binomial)
7	occ_E	E-value for occurrences (binomial)
8	occ_sig	occurrence significance (binomial)
9	rank	rank
10	ovl_occ	number of overlapping occurrences (discarded from the count)
11	forbocc	forbidden positions (to avoid self-overlap)

seq	identifier	exp_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	forbocc
aaaaaa	aaaaaa|tttttt	0.0039539646545	10935	16381.36	0	0e+00	999.00	1	3702	54675
cttttc	cttttc|gaaaag	0.0017351393373	5782	7188.72	1.6e-66	6.7e-63	62.17	2	15	28910
ggatcc	ggatcc|ggatcc	0.0001434041350	230	594.13	1.9e-65	7.8e-62	61.11	3	0	2300
ttaaaa	ttaaaa|ttttaa	0.0019117054886	6482	7920.24	6.8e-63	2.8e-59	58.56	4	543	32410
catgaa	catgaa|ttcatg	0.0010486435958	3394	4344.55	3.8e-51	1.5e-47	46.81	5	247	16970
cttgac	cttgac|gtcaag	0.0004422085347	1337	1832.08	3.2e-34	1.3e-30	29.89	6	0	6685
gatcaa	gatcaa|ttgatc	0.0011548858071	3973	4784.72	6e-34	2.5e-30	29.61	7	245	19865
agcttc	agcttc|gaagct	0.0010722700965	3673	4442.44	6e-33	2.4e-29	28.61	8	126	18365
atatca	atatca|tgatat	0.0011908576718	4143	4933.75	2.7e-31	1.1e-27	26.96	9	194	20715
tataaa	tataaa|tttata	0.0012319046253	4334	5103.81	9.6e-29	3.9e-25	24.40	10	342	21670
aaaatt	aaaatt|aatttt	0.0014280767833	5097	5916.55	4.9e-28	2.0e-24	23.70	11	223	25485
atatat	atatat|atatat	0.0004547677804	1434	1884.11	1.4e-27	5.9e-24	23.23	12	91	14340
agctga	agctga|tcagct	0.0013886969195	5045	5753.40	7.2e-22	3.0e-18	17.53	13	259	25225
tgtcaa	tgtcaa|ttgaca	0.0008065232613	2810	3341.44	1.8e-21	7.2e-18	17.14	14	0	14050
gtacaa	gtacaa|ttgtac	0.0004291427466	1404	1777.95	1.9e-20	7.7e-17	16.12	15	76	7020
cgcgga	cgcgga|tccgcg	0.0003821785924	1247	1583.37	9.2e-19	3.8e-15	14.42	16	67	6235
cctttc	cctttc|gaaagg	0.0007764466766	2738	3216.83	2.4e-18	9.9e-15	14.00	17	6	13690
ctaatc	ctaatc|gattag	0.0002591338295	803	1073.60	3.2e-18	1.3e-14	13.89	18	1	4015
cccccc	cccccc|gggggg	0.0001012307783	255	419.40	3.2e-18	1.3e-14	13.88	19	23	1275
aattat	aattat|ataatt	0.0007415053427	2608	3072.07	4.4e-18	1.8e-14	13.74	20	77	13040
aaattt	aaattt|aaattt	0.0006311269231	2196	2614.77	2e-17	8.2e-14	13.09	21	0	21960
agctgc	agctgc|gcagct	0.0009982041033	3609	4135.58	3e-17	1.2e-13	12.91	22	198	18045
gatcta	gatcta|tagatc	0.0002659860046	837	1101.99	4.4e-17	1.8e-13	12.74	23	11	4185

Figure 6 | Detection of restriction sites in whole genomes. (**a**,**b**) Hexanucleotides having the most significant level of under-representation in the whole genome of (**a**) *Escherichia coli K12* and (**b**) *Bacillus subtilis*, respectively.

Genome-scale detection of rare oligonucleotides

An additional example of application is discussed in **Box 4** (**Table 4**, **Fig. 6**), where we show how to discover restriction sites by detecting under-represented motifs in an entire genome.

Note: Supplementary information is available via the HTML version of this article.

ACKNOWLEDGMENTS This work was supported by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet), by the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265) and by the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture, FRIA (R.J. PhD grant). We acknowledge the students of the Licenciatura en Ciencias Genomicas (CCG-UNAM, Mexico) and the Instituto de Biotecnologia (IBT-UNAM, Mexico) for having tested the protocol and provided useful feedback.

Published online at http://www.natureprotocols.com/
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/

1. Thomas-Chollier, M. *et al.* RSAT: regulatory sequence analysis tools. *Nucleic Acids Res.* **36**, W119–W127 (2008).
2. Brohée, S. *et al.* NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Res.* **Jul 1; 36** (web server issue): w444–51 (2008).
3. Turatsinze, J.-V., Thomas-Chollier, M., Defrance, M. & van Helden, J. Using RSAT to scan genome sequences for transcription factor binding sites and *cis*-regulatory modules. *Nat. Protoc.* doi:10.1038/nprot.2008.97 (2008).
4. Sand, O., Thomas-Chollier, M., Vervisch, E. & van Helden, J. Analyzing multiple data sets by interconnecting RSAT programs via SOAP Web services access—an example with ChIP-chip data. *Nat. Protoc.* doi:10.1038/nprot.2008.99 (2008).
5. Brohée, S., Faust, K., Lima-Mendez, G., Vanderstocken, G. & van Helden, J. Network Analysis Tools: from biological networks to clusters and pathways. *Nat. Protoc.* doi:10.1038/nprot.2008.100 (2008).
6. van Helden, J., André, B. & Collado-Vides, J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* **281**, 827–842 (1998).
7. van Helden, J., Rios, A.F. & Collado-Vides, J. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.* **28**, 1808–1818 (2000).
8. van Helden, J., del Olmo, M. & Pérez-Ortín, J.E. Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals. *Nucleic Acids Res.* **28**, 1000–1010 (2000).
9. Janky, R. & van Helden, J. Evaluation of phylogenetic footprint discovery for predicting bacterial cis-regulatory elements and revealing their evolution. *BMC Bioinformatics* **9**, 37 (2008).
10. Schneider, T.D., Stormo, G.D., Gold, L. & Ehrenfeucht, A. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* **188**, 415–431 (1986).
11. Hertz, G.Z. & Stormo, G.D. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15**, 563–577 (1999).
12. Stormo, G.D. & Hartzell, G.W. III. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci. USA* **86**, 1183–1187 (1989).
13. Hertz, G.Z., Hartzell, G.W. III. & Stormo, G.D. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* **6**, 81–92 (1990).
14. Bailey, T.L. & Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2**, 28–36 (1994).
15. Bailey, T.L. & Elkan, C. The value of prior knowledge in discovering motifs with MEME. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **3**, 21–29 (1995).
16. Lawrence, C.E. *et al.* Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**, 208–214 (1993).
17. Neuwald, A.F., Liu, J.S. & Lawrence, C.E. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci.* **4**, 1618–1632 (1995).
18. Roth, F.P., Hughes, J.D., Estep, P.W. & Church, G.M. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.* **16**, 939–945 (1998).
19. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. & Church, G.M. Systematic determination of genetic network architecture. *Nat. Genet.* **22**, 281–285 (1999).

20. Hughes, J.D., Estep, P.W., Tavazoie, S. & Church, G.M. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.* **296**, 1205–1214 (2000).
21. Thijs, G. *et al.* A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17**, 1113–1122 (2001).
22. Liu, X., Brutlag, D.L. & Liu, J.S. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.* 127–138 (2001).
23. Schbath, S., Prum, B. & de Turckheim, E. Exceptional motifs in different Markov chain models for a statistical analysis of DNA sequences. *J. Comput. Biol.* **2**, 417–437 (1995).
24. Brazma, A., Jonassen, I., Vilo, J. & Ukkonen, E. Predicting gene regulatory elements *in silico* on a genomic scale. *Genome Res.* **8**, 1202–1215 (1998).
25. Brazma, A., Jonassen, I., Eidhammer, I. & Gilbert, D. Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.* **5**, 279–305 (1998).
26. Blanchette, M., Schwikowski, B. & Tompa, M. Algorithms for phylogenetic footprinting. *J. Comput. Biol.* **9**, 211–223 (2002).
27. Blanchette, M., Schwikowski, B. & Tompa, M. An exact algorithm to identify motifs in orthologous sequences from multiple species. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 37–45 (2000).
28. Tompa, M. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 262–271 (1999).
29. Bussemaker, H.J., Li, H. & Siggia, E.D. Regulatory element detection using a probabilistic segmentation model. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 67–74 (2000).
30. Vanet, A., Marsan, L. & Sagot, M.F. Promoter sequences and algorithmical methods for identifying them. *Res. Microbiol.* **150**, 779–799 (1999).
31. DeRisi, J.L., Iyer, V.R. & Brown, P.O. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278**, 680–686 (1997).
32. Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95**, 14863–14868 (1998).
33. Tamayo, P. *et al.* Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* **96**, 2907–2912 (1999).
34. Brazma, A. & Vilo, J. Gene expression data analysis. *FEBS Lett.* **480**, 17–24 (2000).
35. Lee, T.I. *et al.* Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **298**, 799–804 (2002).
36. Harbison, C.T. *et al.* Transcriptional regulatory code of a eukaryotic genome. *Nature* **431**, 99–104 (2004).
37. Molle, V. *et al.* The Spo0A regulon of *Bacillus subtilis*. *Mol. Microbiol.* **50**, 1683–1701 (2003).
38. Salgado, H. *et al.* RegulonDB (version 5.0): *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res.* **34**, D394–D397 (2006).
39. Huerta, A.M., Salgado, H., Thieffry, D. & Collado-Vides, J. RegulonDB: a database on transcriptional regulation in *Escherichia coli*. *Nucleic Acids Res.* **26**, 55–59 (1998).
40. Wasserman, W.W. & Fickett, J.W. Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.* **278**, 167–181 (1998).
41. McGuire, A.M., Hughes, J.D. & Church, G.M. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res.* **10**, 744–757 (2000).
42. McCue, L. *et al.* Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res.* **29**, 774–782 (2001).
43. van Nimwegen, E., Zavolan, M., Rajewsky, N. & Siggia, E.D. Probabilistic clustering of sequences: inferring new bacterial regulons by comparative genomics. *Proc. Natl. Acad. Sci. USA* **99**, 7323–7328 (2002).
44. Kellis, M., Patterson, N., Endrizzi, M., Birren, B. & Lander, E.S. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **423**, 241–254 (2003).
45. Godard, P. *et al.* Effect of 21 different nitrogen sources on global gene expression in the yeast *Saccharomyces cerevisiae*. *Mol. Cell. Biol.* **27**, 3065–3086 (2007).
46. Gonze, D., Pinloche, S., Gascuel, O. & van Helden, J. Discrimination of yeast genes involved in methionine and phosphate metabolism on the basis of upstream motifs. *Bioinformatics* **21**, 3490–3500 (2005).
47. Simonis, N., Wodak, S.J., Cohen, G.N. & van Helden, J. Combining pattern discovery and discriminant analysis to predict gene co-regulation. *Bioinformatics* **20**, 2370–2379 (2004).
48. Simonis, N., van Helden, J., Cohen, G.N. & Wodak, S.J. Transcriptional regulation of protein complexes in yeast. *Genome Biol.* **5**, R33 (2004).
49. Hulzink, R.J. *et al. In silico* identification of putative regulatory sequence elements in the 5′-untranslated region of genes that are expressed during male gametogenesis. *Plant Physiol.* **132**, 75–83 (2003).
50. Aerts, S., van Helden, J., Sand, O. & Hassan, B.A. Fine-tuning enhancer models to predict transcriptional targets across multiple genomes. *PLoS ONE* **2**, e1115 (2007).
51. Stark, A. *et al.* Discovery of functional elements in 12 Drosophila genomes using evolutionary signatures. *Nature* **450**, 219–232 (2007).
52. Strauch, M., Webb, V., Spiegelman, G. & Hoch, J.A. The Spo0A protein of *Bacillus subtilis* is a repressor of the abrB gene. *Proc. Natl. Acad. Sci. USA* **87**, 1801–1805 (1990).
53. Baldus, J.M., Green, B.D., Youngman, P. & Moran, C.P. Jr. Phosphorylation of *Bacillus subtilis* transcription factor Spo0A stimulates transcription from the spoIIG promoter by enhancing binding to weak 0A boxes. *J. Bacteriol.* **176**, 296–306 (1994).
54. Sierro, N., Makita, Y., de Hoon, M. & Nakai, K. DBTBS: a database of transcriptional regulation in *Bacillus subtilis* containing upstream intergenic conservation information. *Nucleic Acids Res.* **36**, D93–D96 (2007).
55. Roberts, R.J., Vincze, T., Posfai, J. & Macelis, D. REBASE—enzymes and genes for DNA restriction and modification. *Nucleic Acids Res.* **35**, D269–D270 (2007).
56. Roberts, R.J. & Macelis, D. REBASE—restriction enzymes and methylases. *Nucleic Acids Res.* **28**, 306–307 (2000).
57. Kurtz, S. *et al.* REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res.* **29**, 4633–4642 (2001).
58. Kurtz, S. & Schleiermacher, C. REPuter: fast computation of maximal repeats in complete genomes. *Bioinformatics* **15**, 426–427 (1999).
59. Robin, S., Rodolphe, F. & Schbath, S. *DNA, Words and Models—Statistics of Exceptional Words* (Cambridge University Press, Cambridge, 2005).
60. Nuel, G. & Prum, B. *Analyse statistique des séquences biologiques: modélisation markovienne, alignements et motifs* (Hermes Science Publishing, London, England, 2007).
61. Brazma, A., Vilo, J., Ukkonen, E. & Valtonen, K. Data mining for regulatory elements in yeast genome. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **5**, 65–74 (1997).
62. Sinha, S. & Tompa, M. YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.* **31**, 3586–3588 (2003).
63. Reinert, G. & Schbath, S. Compound Poisson and Poisson process approximations for occurrences of multiple words in Markov chains. *J. Comput. Biol.* **5**, 223–253 (1998).
64. El Karoui, M., Biaudet, V., Schbath, S. & Gruss, A. Characteristics of Chi distribution on different bacterial genomes. *Res. Microbiol.* **150**, 579–587 (1999).
65. Vandenbogaert, M. & Makeev, V. Analysis of bacterial RM-systems through genome-scale analysis and related taxonomy issues. *In Silico Biol.* **3**, 127–143 (2003).