Using RSAT to scan genome sequences for transcription factor binding sites and *cis*-regulatory modules

Jean-Valery Turatsinze[1,2], Morgane Thomas-Chollier[1,2], Matthieu Defrance[1] & Jacques van Helden[1]

[1]Laboratoire de Bioinformatique des Génomes et des Réseaux (BiGRe), Université Libre de Bruxelles CP 263, Campus Plaine, Boulevard du Triomphe, B-1050 Bruxelles, Belgium. [2]These authors contributed equally to this work. Correspondence should be addressed to J.v.H. (jacques.van.helden@ulb.ac.be).

Published online 18 September 2008; doi:10.1038/nprot.2008.97

This protocol shows how to detect putative *cis*-regulatory elements and regions enriched in such elements with the regulatory sequence analysis tools (RSAT) web server (http://rsat.ulb.ac.be/rsat/). The approach applies to known transcription factors, whose binding specificity is represented by position-specific scoring matrices, using the program *matrix-scan*. The detection of individual binding sites is known to return many false predictions. However, results can be strongly improved by estimating *P* value, and by searching for combinations of sites (homotypic and heterotypic models). We illustrate the detection of sites and enriched regions with a study case, the upstream sequence of the *Drosophila melanogaster* gene *even-skipped*. This protocol is also tested on random control sequences to evaluate the reliability of the predictions. Each task requires a few minutes of computation time on the server. The complete protocol can be executed in about one hour.

INTRODUCTION

The coordination of biological processes relies on a tight regulation of gene expression depending on time, tissues and cell types. Transcription factors regulate gene expression by binding at specific locations on DNA, called transcription factor binding sites (TFBSs). For example, at least 44 binding sites have been characterized for the transcription factor *Kruppel* in the genome of the fruit fly *Drosophila melanogaster* (a subset of these sites is shown in **Fig. 1a**).

Specific bioinformatics approaches have been developed to identify TFBSs in DNA sequences (reviewed in ref. 1). *Pattern matching* consists in searching for sites recognized by a known transcription factor, and requires prior knowledge of the motif that describes the binding specificity of this transcription factor. *Pattern discovery* consists in predicting novel motifs from a set of sequences that are putatively coregulated by some transcription factor, without any prior information about their binding specificity.

Regulatory sequence analysis tools

The regulatory sequences analysis tools (RSAT)[2,3] are a suite of specialized programs for detecting regulatory elements. The website integrates a set of modular tools that can be combined to perform all the steps from sequence retrieval to drawing of graphical maps displaying the predicted sites, including several methods for pattern discovery and pattern matching.

This is the first article in a series of four protocols for the analysis of regulatory sequences (http://rsat.ulb.ac.be/rsat/) and biological networks (http://rsat.ulb.ac.be/neat/). In this protocol, we present a pattern-matching procedure for predicting putative TFBSs as well as enhancer regions (*cis*-regulatory element-enriched regions, CRERs) using the program *matrix-scan*. The second article[4] describes a protocol for the *ab initio* discovery of biological signals in biological sequences (pattern discovery). The third article[5] shows how RSAT can be queried through a programmatic interface, to automate the analysis of multiple data sets (e.g., coexpression clusters). The fourth article[6] describes a workflow for deciphering biological networks by combining network comparison, module identification and path finding.

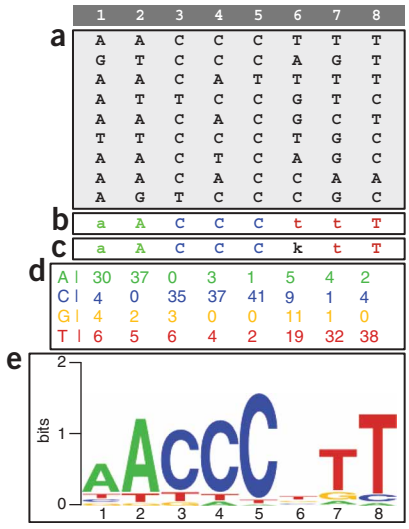

Figure 1 | Representations of the binding specificity for the Krüppel transcription factor of *Drosophila melanogaster*. (**a**) A subset of the collection of 44 Krüppel sites from ORegAnno database, aligned using the program MEME. Panels (**b**–**e**) are based on the whole collection of Krüppel sites. (**b**) Strict consensus of the selected sites. (**c**) Degenerate consensus. (**d**) Position-specific scoring matrix (PSSM) obtained using *convert-matrix*. Each column of the matrix represents one position of the motif and the numbers indicate the nucleotide absolute frequencies at this position of the aligned sites. (**e**) Sequence Logo obtained using WebLogo (http://weblogo.berkeley.edu/logo.cgi). Each column represents one position of the motif, and the letters indicate which nucleotides are found at a given position. The total height of each column reflects its information content, that is, how far it discards from the background nucleotide frequencies. The height of each letter is proportional to the frequency of the corresponding nucleotide at the given position.

BOX 1 | MATRIX SCORING SCHEME

Position-specific scoring matrices (PSSMs) are motif descriptors that take into account the residue frequencies at each position of the motif. They are widely used in pattern matching and pattern discovery programs, and varying file formats have been proposed. Many PSSM formats are supported in regulatory sequence analysis tool (RSAT) and the program *convert-matrix* can be used to perform interconversions between these formats. **Figure 1** illustrates the way to build a PSSM (**Fig. 1d**) from a collection of aligned sites (**Fig. 1a**). A graphical representation of the motif as a sequence logo is helpful to obtain a global view of the motif (**Fig. 1e**).

The program *matrix-scan* scans the input sequences with a PSSM by selecting, at each position, a sequence segment of the same length as the motif and by assigning a score to this segment. The 'matches', that is, sequence segments scoring above some predefined threshold, are considered as putative binding sites for the studied transcription factor. The weight score is derived from the theory developed by Jerry Hertz and Gary Stormo[12], upgraded with the capability to use Markov chain-based background models, as proposed by Thijs *et al.*[14].

The weight of a sequence segment (W_s) is computed as the log-ratio between two probabilities.

$$W_s = \log\left(\frac{P(S|M)}{P(S|B)}\right)$$

In this formula, $P(S|M)$ is the probability for the sequence segment S to occur according to the motif model M (the PSSM), whereas $P(S|B)$ is the probability for the same sequence S to occur under the background model B. The weight score is thus the log-ratio of the likelihood of S in these two respective models. Positive weight scores indicate that a sequence segment is more likely to be an instance of the motif than an instance of the background (the genomic context).

Pattern matching methods

Pattern matching methods can be divided into two broad categories, *string-based* and *matrix-based*, depending on how the motif is represented. In string-based approaches, a simple string that summarizes the collection of binding sites (**Fig. 1a**) represents the specificity of a transcription factor. This consensus can either be described with an alphabet restricted to the 4-letter code (A, C, G and T; strict consensus, **Fig. 1b**), or with the 15-letter International Union of Pure and Applied Chemistry (IUPAC) code for representing ambiguous positions (degenerate consensus, **Fig. 1c**).

Position-specific scoring matrices (PSSMs) offer a more expressive description of the binding specificity, taking into account the frequency of each nucleotide at each position of the motif (**Fig. 1d**). The sequence logo (**Fig. 1e**) provides an intuitive graphical representation of the relative importance of each residue at each position of the motif.

Specialized databases such as TRANSFAC[7,8] and RegulonDB[9,10] hold collections of PSSMs built from experimentally verified binding sites. Various matrix-based programs have been developed to predict individual binding sites (e.g., patser[11,12], MotifLocator[13,14], MATCH[15]) and putative *cis*-regulatory modules (CRMs) (ClusterBuster[16], Modulefinder[17], Target Explorer[18], TFBScluster[19,20], eCis-analyst[21,22], CisPlusFinder[23], ModuleSearcher[24]). Cross-species conservation is also used to refine the predictions of individual binding sites, as in the rVISTA program[25].

CRMs are combinations of sites located in a delimited region, where several transcription factors interact to regulate the expression of a gene. In RSAT, the program *matrix-scan* detects both individual sites and putative CRMs referred hereafter as CRERs. CRERs are defined as short sequence regions with a significant high density of predicted sites. CRERs can be considered as computer-based predictions of CRMs. The program *matrix-scan* supports the prediction of homotypic modules (i.e., enrichment in binding sites for a single transcription factor) or heterotypic modules (i.e., enrichment in binding sites for multiple transcription factors).

Pattern-matching methods are known to return many false positive predictions that hamper the interpretation of the results. Efforts have consequently been focused on developing approaches that reduce false positive predictions. When a sequence is scanned with a pattern-matching program, a score is assigned to each position of this sequence according to its similarity with the motif described by the PSSM (see **Box 1** for details on the scoring scheme implemented in *matrix-scan*). The predictions that reach some predefined threshold are retained as predicted binding sites. It is crucial to determine a threshold that ensures a reasonably low rate of false positives, while maintaining a sufficient rate of correct predictions. However, this score threshold is too commonly set to some arbitrary value. In addition, the expected distribution of scores varies depending on the matrices (see **Box 2** for details on expected distributions), so that a given weight score might be meaningful for some matrix but not for some others. An important advantage of *matrix-scan* is that it estimates the *P* value of each site. The *P*-value is the probability to obtain a given score by chance, and can be used to estimate the probability of mistakenly considering a hit as significant. The threshold can thereby directly be interpreted as a risk of false positive, which makes it more informative than the usual weight scores.

Functional *cis*-regulatory elements are often found densely packed in regions called CRMs[16,21,26]. This observation is very useful for the detection of binding sites with bioinformatics methods. Indeed, predicted sites that are located within CRERs are more likely to be correct than isolated predictions. *Matrix-scan* is able to detect regions showing a higher density of sites than expected by chance. A *P*-value is calculated for each CRER using the binomial formula[27] and a threshold can be applied on this value to return significant CRERs only. Combining the detection of individual sites and CRERs consequently contributes to the reduction of false predictions.

The other pattern matching programs focus on the prediction of either individual binding sites or CRMs. Regarding individual binding sites, patser is efficient and computes *P*-values, but it only supports one type of background model (Bernoulli). In contrast, MotifLocator supports Markovian background models, but it does not return *P*-values for the predicted sites. With respect to the prediction of CRMs, ClusterBuster and Target Explorer are limited to Bernoulli background models and are restricted to a

BOX 2 | SITE *P*-VALUES

The primary scores assigned by *matrix-scan* are weights that are calculated from a position-specific scoring matrix (PSSM) and a background model (see **Box 1** for details on the scoring scheme). For a given PSSM/background model combination, the list of all possible weight values corresponding to all possible sequence segments can be calculated. As some weights can be generated by several sequence segments, they are more frequent than the other weights. These expected frequencies are calculated to obtain the expected distribution of all the scores. An efficient algorithm has been proposed for Bernoulli background models (equivalent to Markov models of order 0)[39]. The program *matrix-distrib* extends this algorithm to higher order Markov models and computes the complete expected distribution of scores for a given PSSM and a given background model. This expected distribution of scores is used to estimate the *P*-value of the weights.

For the prediction of binding sites, the program *matrix-scan* supports thresholds on various fields, including the weight score W_s and the site *P*-value. For scanning sequences, we recommend the threshold on *P*-value, which gives a better intuition about the risk associated to each prediction. For example, with a *P*-value threshold of 0.001, one false positive prediction is expected every kilobase, whereas with a *P*-value of 0.0001, one false positive prediction is expected every 10 kb. Setting the threshold on the *P*-value rather than the weight score is even more crucial when sequences are scanned with multiple matrices. Indeed, each motif has its own size and information content, which critically influences the expected distribution of weights. A weight of six may thus be associated to a very low probability for a given matrix (and thus return reliable sites) while having a higher probability for another matrix (and thus return many false predictions).

predefined number of matrices. Modulefinder and CisPlusFinder are dedicated to cross-species detection of CRMs. Compared to these programs, the advantages and limitations of matrix-scan are described below.

Main advantages of *matrix-scan*

(1) The program supports the prediction of individual sites as well as CRERs, for either homotypic or heterotypic models.
(2) Multiple PSSMs can be treated in a single run of the program, facilitating the handling of the results (a single file with all the matches) and their display on a feature map. The treatment of multiple PSSMs is essential to support heterotypic models for predicting CRMs.
(3) Matrix-scan integrates three types of background model estimation, including an original adaptive model (progressive update of the model during the scanning, according to the local context).
(4) Thresholds can be set independently on each supported score, including the *P*-value, which can readily be interpreted in terms of risk of false positives. *P*-values are calculated for Markov chain-based background models of any order.
(5) Many formats are supported for the description of PSSM and background model.

Main limitations of *matrix-scan*

(1) Matrix-scan is written in Perl and the computing time is thus slower than for some compiled programs. On the web server, the average scanning speed is 0.4 s per kb for a single-matrix search, and this time increases linearly with the sequence length and the number of matrices. For genome-wide analysis, we recommend to use the email output. For people having basic programming skills, the tool can also be used through the web services interface rather than on the web server (see protocol[5]), or as a stand-alone application.
(2) Like all pattern matching programs, *matrix-scan* requires one to provide matrices, and thus to start from some prior knowledge about the binding specificity of one or several transcription factors. For *de novo* prediction of binding sites, pattern discovery methods should be considered (see protocol[4]).
(3) Presently, *matrix-scan* does not support cross-species comparisons to prioritize binding site predictions. The RSAT website, however, includes the necessary modular tools to perform this task: orthologous genes can be selected with *get-orthologs*, and their promoters can be obtained with *retrieve-seq*.

Study case: regulatory region of the *Drosophila even-skipped* gene

In this protocol, we show how to use PSSMs to predict TFBSs and CRERs. We discuss ways to interpret the results, evaluate the risk of false positive predictions and prioritize the predictions for further experimental validations. The RSAT website facilitates these analyses by providing a wide collection of tools for sequence retrieval, analysis, graphical representation and format conversion through interconnected web pages. **Figure 2** illustrates the different steps of the procedure and the links between the programs. Note that this workflow is completely modular, so that users can enter at any of the steps with their own data.

This protocol is illustrated by a study case based on the *even-skipped* (*eve*) gene in *D. melanogaster*. *even-skipped* is a pair-rule gene that codes for a homeobox transcription factor and plays a critical role in the formation of the antero-posterior axis during embryogenesis[28]. Its expression in stripes during embryonic development is tightly regulated by transcription factors that bind on *cis*-regulatory elements located in its upstream region. This region has been intensively annotated in ORegAnno[29,30] and REDfly[31] with experimentally verified binding sites and modules, respectively. This annotation will serve as a reference set for evaluating the predictions. We will retrieve a 5,500-bp region located upstream of the *eve* coding gene, and scan this region with PSSMs corresponding to 12 known *even-skipped* regulators. The first part of this protocol illustrates the prediction of individual TFBSs, and the second part illustrates the prediction of heterotypic modules in the *even-skipped* promoter.

MATERIALS

EQUIPMENT

- A personal computer with connection to Internet and a web browser
- A collection of PSSMs that represent transcription factor binding motifs (see EQUIPMENT SETUP)
- DNA sequences to be scanned provided as a text file (see Step 11 for supported file formats). Alternatively, sequences may be fetched directly from RSAT (Steps 1–9 of the PROCEDURE). A list of gene names or identifiers is then required, such as those defined in the GenBank[32] (http://www.ncbi.nlm.nih.gov/) or Ensembl[33] (http://www.ensembl.org) databases

EQUIPMENT SETUP

Collection of PSSMs The PSSMs are described in text files (see Step 10 for supported file formats). Ready-to-use PSSMs can be obtained from specialized databases such as TRANSFAC[7,8] (http://www.gene-regulation.com/pub/databases.html), JASPAR[34,35] (http://jaspar.genereg.net/) or RegulonDB[9,10] (http://regulondb.ccg.unam.mx). Users can also build their own matrices from a collection of experimentally validated TFBS sequences obtained from either the literature or from databases dedicated to annotated TFBSs such as ORegAnno[29,30] (http://www.oreganno.org). Programs such as MEME[36] (http://meme.sdsc.edu/meme/) or consensus[11] (http://rsat.ulb.ac.be/rsat/) can be used to align these TFBS sequences and produce the PSSM.
A collection of PSSMs built from ORegAnno TFBSs is available in RSAT (http://rsat.ulb.ac.be/rsat/data/motifs/Drosophila_melanogaster/ORegAnno/).

The program *convert-matrix* included in RSAT performs interconversions between a variety of PSSM file formats (AlignACE, cluster-buster, clustal, consensus, gibbs, MEME, MotifSampler, TRANSFAC).

Collection of *Drosophila* PSSM for testing this protocol We converted collections of *Drosophila* binding sites from ORegAnno into PSSMs, and placed them on the RSAT website, in the supplementary material section (http://rsat.ulb.ac.be/rsat/data/published_data/nature_protocols/matrix_scanning/). Before starting the protocol, open a connection to this site in a separate window of the web browser. At several steps of the protocol, you will come back to this site to pick up some sample data sets.

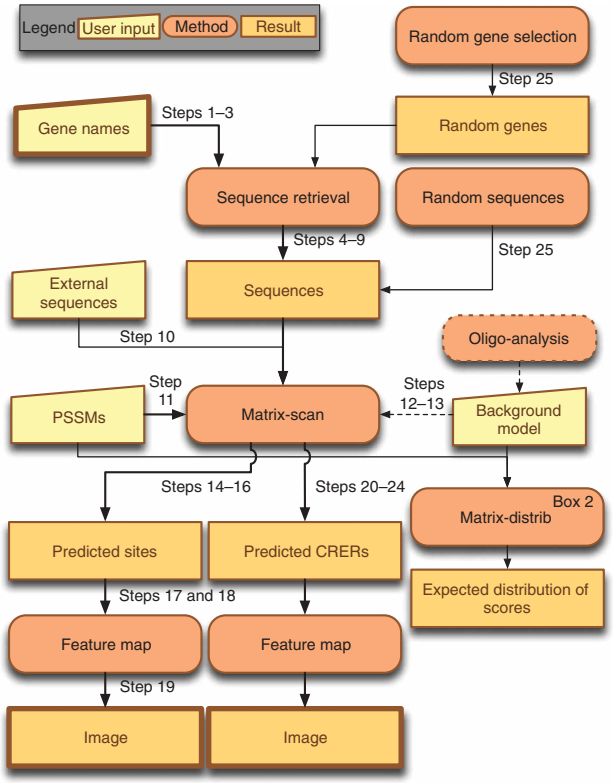

Figure 2 | Graphical flowchart showing the links between regulatory sequence analysis tool (RSAT) programs used in the protocol. Rounded rectangles indicate tools, trapezoids correspond to user-provided input and rectangles represent results.

PROCEDURE

Sequence retrieval

1| This section (Steps 1–9) is not necessary for users who already have their own sequences to analyze. In a web browser, open a connection to the RSAT web server (http://rsat.ulb.ac.be/rsat/). The main RSAT server is located in Belgium but several mirror servers are available from the main page. Click on the `Sequence retrieval` title in the left menu. A sub-menu opens. Click on the link `retrieve sequence` to open the 'retrieve sequence' form. **Figure 3** illustrates the filled-in form for the *even-skipped* example.

2| In the menu at the top of the form, select the organism (e.g., *Drosophila melanogaster* for this illustration).
? TROUBLESHOOTING

3| In the `genes` section, copy a list of gene identifiers in the box, separated by carriage returns. Only the first word of each line is considered as a query. To upload a list of gene identifiers from a file, click on the browse button and choose the appropriate file on your computer. For the study case discussed in this protocol, simply type 'eve'.

4| To retrieve the promoter sequences of the chosen genes, choose 'mRNA' as `Feature type`, and select 'upstream' as `Sequence type`. The transcription start site (TSS) is then considered as the origin (position 0) of the retrieved sequence.
▲ CRITICAL STEP The option `Feature type` indicates the annotation type that will serve as reference for specifying the positions relative to the gene. In general, we would like to use the TSS as origin (position 0), and we would thus select mRNA annotations. However, these annotations are missing in the majority of the microbial genomes. To work with a genome for which no mRNA is available, choose 'CDS' as `feature type` and upstream sequences will be retrieved relative to the start codon.

5| Specify the sequence limits in the `from` and `to` boxes. The regions upstream of the origin (TSS in our study case) are specified by negative values. For further details on this coordinate system, refer to the tutorial of the tool (follow the `tutorial` link at the bottom of the page). For the *eve* study case, select the upstream region from −5500 to −1.

6| Select `Prevent overlap with neighbour genes` to limit the upstream sequences when a predicted open reading frame is located within the range defined by the option `from`.
▲ CRITICAL STEP When the option `Prevent overlap with upstream genes` is active, the sequence size is adapted to discard the coding sequences of neighbor genes. The motivation is that most *cis*-acting elements are located in the non-coding regions. Thus, if the neighbor gene is too close and overlaps with the region to be retrieved, the included coding sequence is likely

to add noise without bringing any additional signal. More important, coding and non-coding sequences generally have very different background models. The inclusion of coding sequences from neighbor genes may thus lead to biases in the estimations of *P*-values. For bacterial sequences, it is essential to prevent overlap with upstream genes, because intergenic distances are often very short, particularly between pairs of genes comprised in the same operon.

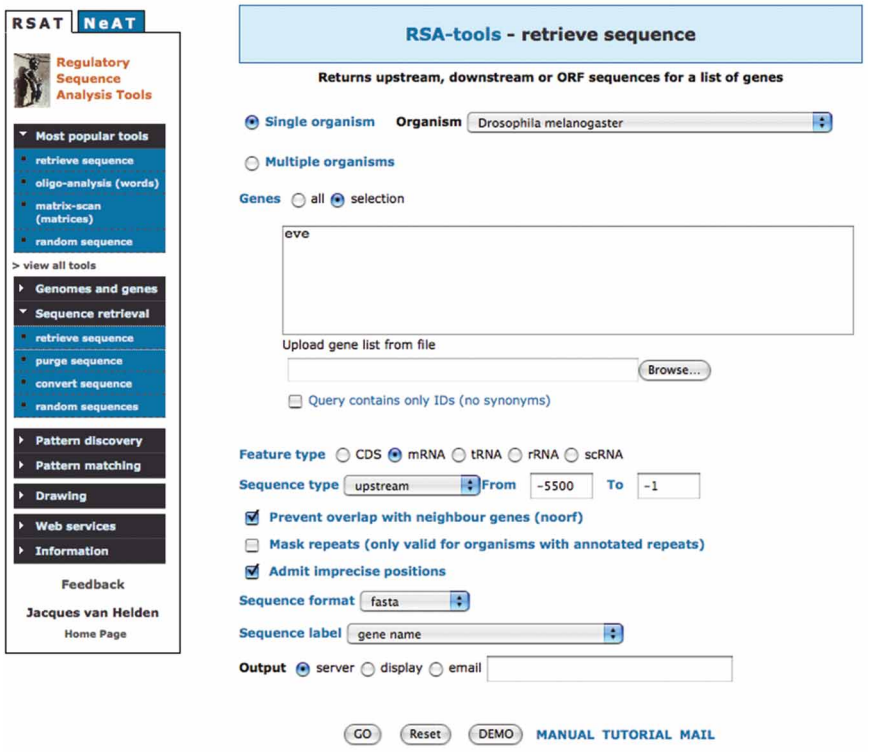

Figure 3 | The retrieve sequence form, filled-in for the *even-skipped* example.

7| Select `Mask repeats` to use the genome version where repeats are masked (i.e., replaced by 'N' characters). This option is valid only for organisms with annotated repeats (refer to the list of organisms for which this option is valid). If this option does not apply for the organism of interest, the user is invited to mask his/her sequences with the program RepeatMasker (http://www.repeatmasker.org/). For the example, leave this option unchecked. ▲ CRITICAL STEP For vertebrate genomes, the presence of repetitive elements hampers the detection of motifs, because these repetitive sequences have a very distinct composition than the rest of the genome.

8| For the option `Output`, keep 'server' checked. Click on the `GO` button to retrieve the sequence. ? TROUBLESHOOTING

9| After a few seconds, the result page should appear. In the section Result, there is a link to the text file containing the retrieved sequences (**Fig. 4**). Optionally, click on this link to see the actual sequences, but this is not necessary: the sequences are stored on the server for further analysis and can be used without transferring them back and forth over the web. At the bottom of the page, a `Next step` section allows to transmit the retrieved sequence directly as input to another RSAT program. RSAT actually offers a variety of alternative programs for pattern discovery and pattern matching. Click on the `matrix-scan` button to load the retrieved sequences automatically in the 'matrix-scan' form (**Fig. 5**). ? TROUBLESHOOTING

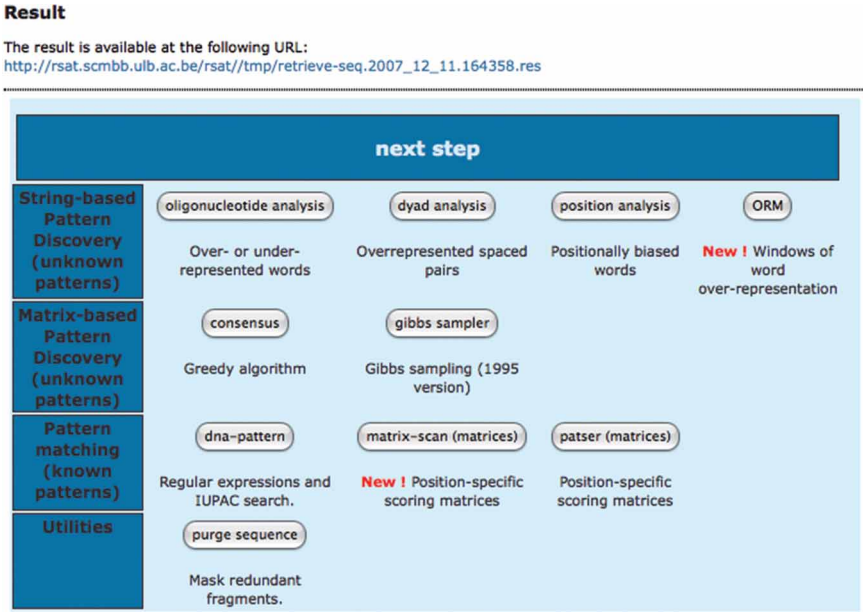

Figure 4 | The retrieve sequence result page, displaying a button to send the sequences to matrix-scan, in the `Next step` section.

Prediction of individual TFBSs

10| After having followed Steps 1–9, the `Sequence` section of the form should indicate `transferred from previous query`. Alternatively, you may directly enter your own sequences in the 'matrix-scan' form (**Fig. 5**). For this, you need to obtain an empty form by clicking on the title `Pattern matching` of the left menu and select the `matrix-scan` link. For a quick

test, the 'matrix-scan' form already contains DEMO buttons: 'DEMO 1' will fill in the form with the study case parameters to return individual sites. The sequences to be scanned can be specified either by pasting the sequence in the box, or by uploading a sequence file with the browse button (this is required for large sequence files). Choose the format corresponding to your sequences with the pop-up menu Format.

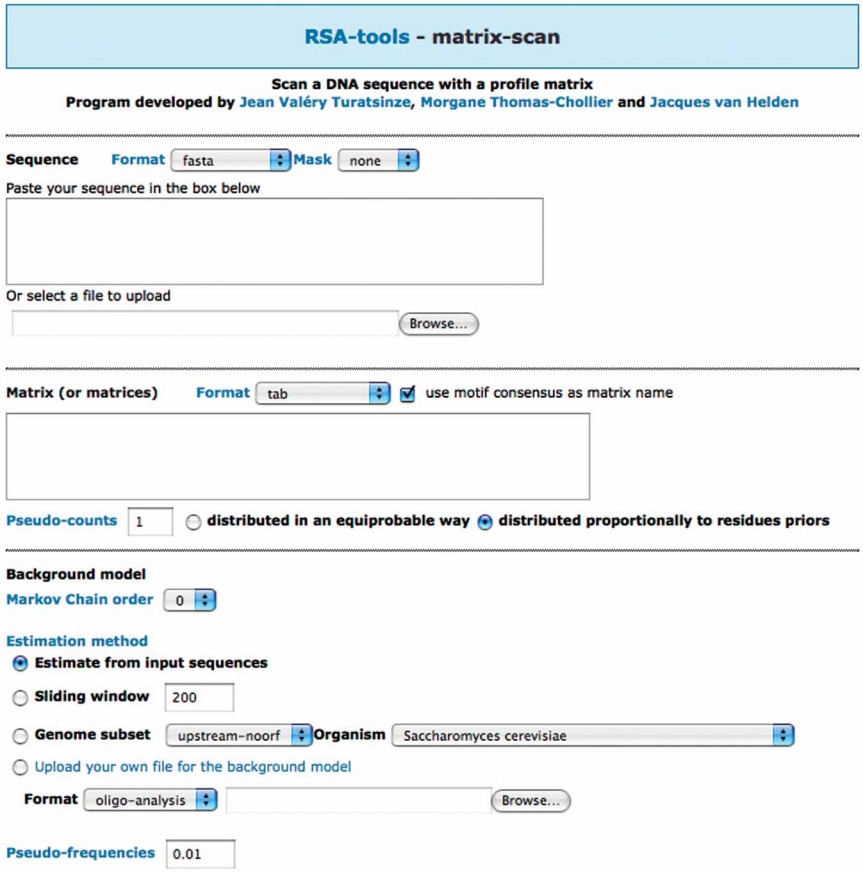

Figure 5 | The matrix-scan form: Sequence, Matrix and Background model sections.

11| The Matrix section allows specifying the transcription factor-binding motif(s). In the default format (tab), several matrices can be provided (matrices are separated by a line containing a double slash '//'). For the *even-skipped* study case, go to the supplementary material website specified above, and copy/paste the content of the file *oreganno_eve_12_matrices.txt* in the Matrix box. In the menu Matrix format, select 'transfac'. A wide range of PSSM formats are supported in RSAT and can be specified using the pop-up menu format. In the example, the matrices are provided in TRANSFAC format that includes the matrix names. Depending on the format, the PSSM name may not be contained in the PSSM specification. If the selected format does not support a matrix name, you can optionally check the option use motif consensus as matrix name to compute automatically, for each PSSM, a degenerate consensus that will be used as matrix name. With the matrices provided for the study case, leave this option unchecked because the TRANSFAC format supports a name for each matrix. To reduce the bias that arises from the small number of TFBSs used to construct the matrices, pseudo-counts are added in each cell of the matrix. In the example, set pseudo-counts to 1 and select distributed proportionally to residues priors.

12| The next section of the form provides several options for specifying the background model (the statistical model for the sequences that do not correspond to instances of the motif). The choice of the background model crucially affects the results (see **Box 3** for details on background models). For first analysis, select Markov chain order 0. After executing the protocol, you can optionally come back to this step and explore the effect of the Markov order on results.

! CAUTION Increasing the Markov chain order extends the computing time of the *P*-value, particularly for the adaptive models (option 'sliding window'). In any case, with the option 'sliding window', one should select a model of very low order (0 or 1) to avoid overfitting (as discussed in **Box 3**). High-order Markov models calibrated on genome subsets (e.g., upstream-noorf) are suitable, but the precomputation of the *P*-value distribution will impose a delay of a few seconds for each input matrix.

13| The option estimation method presents four alternative ways to train the background model (see **Box 3** for explanations). For the *eve* study case, check the option 'Genome subset', and select 'upstream-noorf' and '*Drosophila melanogaster*'.

14| The section Scanning options determines the scanning mode and the parameters to return (**Fig. 6**). The menu Search strands indicates whether sequences should be scanned on a single strand or on both strands. Because most eukaryotic transcription factors act in a strand-insensitive way, it is generally recommended to leave this option on 'both strands'. The selector Origin specifies whether the origin for reporting coordinates should be the end or the start of the sequences. By default, the end is considered as the origin, so that the hits are reported with negative coordinates for upstream sequences. Finally, the option decimals value specifies the number of decimals to be displayed for the weight scores. In the example, set score decimals to 1.

! CAUTION Increasing the number of decimals significantly extends the time and memory usage for computing the *P*-value. This is particularly true for higher order Markov models. In addition, it makes not much sense to compute weight scores with a high number of decimals, because there is generally an intrinsic limitation of their precision due to the restricted number of sites used to build PSSMs.

BOX 3 | BACKGROUND MODELS

As described in **Box 1**, the weight score (W_s) is based on the log-ratio between two probabilities: the probability for a sequence segment to be an instance of the motif $P(S|M)$ versus its probability to occur given the background model $P(S|B)$. The choice of an appropriate background model thus crucially affects transcription factor binding site predictions. The program *matrix-scan* supports Bernoulli models that assume independence between successive residues, as well as higher order Markov chains, where the probability to find a residue at a given position depends on the residues found at the m preceding positions (m is the order of the Markov chain). Markov models of order m determine the frequencies of words of length $k = m + 1$. A Markov chain of order 0 corresponds to a Bernoulli model. A general description of Markov chain models is beyond the scope of this protocol, but can be found in many textbooks on sequence analysis, for example in ref. 40.

The *matrix-scan* form allows you to choose among various ways to train the background model.

1. With the first method `Estimate from input sequences`, the background model is estimated on the basis of all the sequences provided as input, before starting the scanning.
2. With the method `Sliding window`, an adaptive background model is computed during the scanning, to account for local variations in genome composition (e.g., CpG islands). The size of the sliding window can be specified in the box.

! CAUTION The background estimation `Sliding window` increases computing time, as the model is updated for every scored segment.

3. The next method `Genome subset` is based on organism-specific precalibrated background models.
4. To use a model trained on a custom sequence set, you can provide a precalculated background model file in the `upload your own file` section. Several background model formats are supported to ensure compatibility with other programs (*oligo-analysis*, MEME, MotifSampler).

Higher order Markov chains require sequences with a sufficient size for their training. A background model of higher order trained with a too short sequence will provoke an overfitting of the model, and no sites will be detected anymore. We thus recommend using a very small Markov order (0 or 1) when the model is trained from the input sequences (and *a fortiori* with the option `Sliding windows`), and higher order models with genome-scale calibrations.

15| *Matrix-scan* offers three complementary types of analyses that can be selected at the top of the section `Return`: 'individual matches', 'CRERs' or 'statistics about site over-representation in the whole set of input sequences', respectively. For each return type, the form includes a set of user-selectable return fields and thresholds. In this first part of the protocol, we will predict individual matches of the PSSM. Select `Individual matches` as return type. Defining a threshold on the *P*-value is the preferred approach (see **Box 2** for more details on *P*-values). When the background model is computed in an adaptive way with the 'sliding window' option, the threshold should be put on the score for computational reasons. For the *eve* study case, set the `Upper threshold` on '*P*-value' to 0.0001.

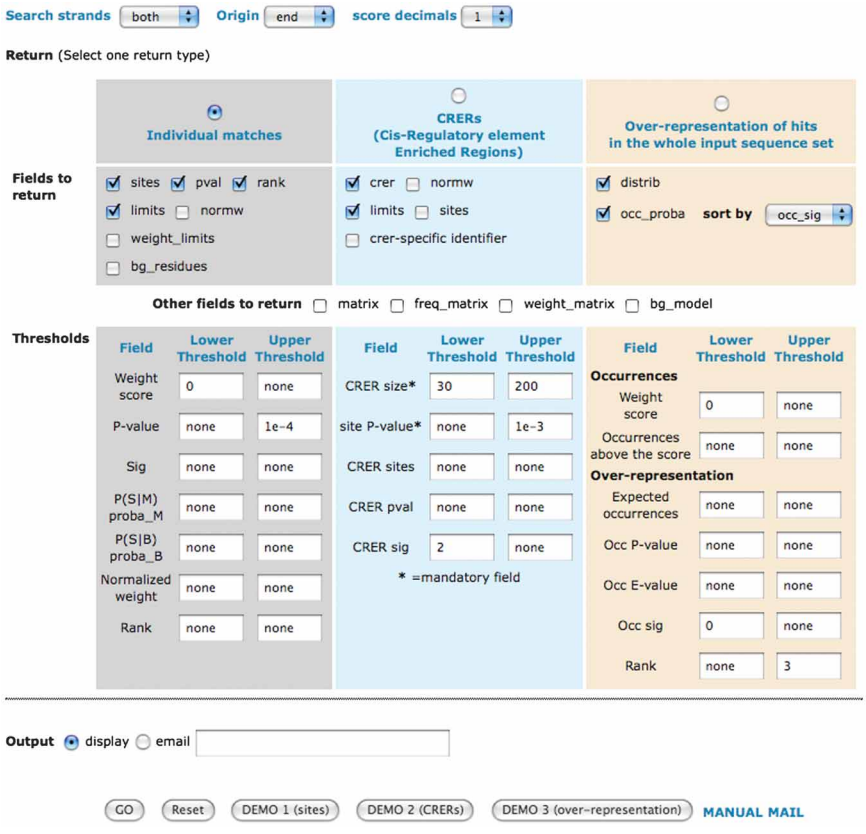

Figure 6 | The matrix-scan form: Scanning options and Return sections.

! CAUTION The `rank` option increases memory usage, as all matches are held in memory to be finally sorted.

16| For the `output`, keep the option 'display' checked. For large data sets or computer-intensive options, select the 'email' output and fill in a valid address. Click on the GO button to scan the sequence. After a few seconds, a window should appear with the predicted sites (**Fig. 7**).

? TROUBLESHOOTING

Graphical presentation of the putative sites and CRERs

17| The *matrix-scan* results are displayed in tabular files containing detailed information on the location of the predicted sites and on their scores. To facilitate the interpretation of the results, matches can be displayed along the scanned sequence on a graphical feature map. At the bottom of the matrix-scan result page, a section `Next step` allows one to send the matrix-scan results directly as input to another RSAT program. Click on the `feature-map` button to open the 'feature-map' form.

18| A title for the analysis can be specified in the `title` box. For a first trial, leave the other options to their default values.

19| To produce an image that can be displayed in your browser, set the option `image format` to 'png'. Alternatively, to produce high-quality images, set the option `image format` to 'ps'. Click on the `GO` button to generate the image. After a few seconds, the feature-map (as illustrated in **Fig. 8**) should be displayed in your browser.

Prediction of CRERs

20| The second type of analysis supported by matrix-scan permits one to detect regions of a few hundred residues that have a higher density of matches than expected by chance. Come back to the `Return` section of the 'matrix-scan' form (Step 16). For a quick test, click on the 'DEMO 2' button to fill in the form with the study case parameters for the CRER output.

21| To search for CRERs, select 'CRERs' as `return type`. It is mandatory to specify the thresholds for `crer_size` and `site P-value`. For the *even-skipped* study case, set the 'Upper threshold' for `CRER size` to 200 and the 'Lower threshold' for `CRER size` to 30.

```
matrix-scan  -v 1 -matrix_format transfac -m tmp/matrix-scan.2007_12_21.113021.matrix -pseudo 1 -decimals 1 -2str -origin -0 -bgfi
Input files
       input   tmp/matrix-scan.2007_12_21.113021.seq
       bg      data/genomes/Drosophila_melanogaster/oligo-frequencies/1nt_upstream-noorf_Drosophila_melanogaster-ovlp-1str.freq
Matrix files
       matrix 1tmp/matrix-scan.2007_12_21.113021.matrix
Sequence format         fasta
Pseudo counts           1
Background model
       Method           file
       Bernoulli model (order=0)
       Strand           sensitive
       Background pseudo-frequency      0.01
       Residue probabilities
              a         0.30414
              c         0.20046
              g         0.19860
              t         0.29680
Thresholds     lower    upper
       pval            NA       0.0001
       score           0        NA
```

seq_id	ft_type	ft_name	strand	start	end	sequence	weight	proba_M	proba_B	Pval	ln_Pval	sig
eve	limit	SEQ_START	DR	-5500	-5500	.						
eve	limit	SEQ_END	DR	-1	-1	.						
eve	site	eve	R	-5257	-5243	CAGCAAATTCGCGCT	7.2	8.2e-07	5.7e-10	9.3e-05	-9.280	4.030
eve	site	ttk	D	-5221	-5215	CAGGACC	8.4	1.2e-01	2.9e-05	2.9e-05	-10.435	4.532
eve	site	prd	R	-5010	-5003	CACCGCAC	7.3	8.7e-03	5.9e-06	9.2e-05	-9.291	4.035
						• • •						
eve	site	prd	D	-63	-56	CACCGCAC	7.3	8.7e-03	5.9e-06	9.2e-05	-9.291	4.035
eve	site	eve	D	-60	-46	CGCACGATTAGCACC	12.7	1.4e-04	3.8e-10	4.3e-08	-16.958	7.365

```
Matrices
       matrix  name    ncol    nrow    pseudo  Pmin    Pmax    Wmin    Wmax    Wrange  prior
       1       Kr      8       4       1       1.6e-15 0.086   -22.700 8.600   31.300  a:0.304 c:0.200 g:0.199 t:0.297
       2       Med     6       4       1       1.4e-10  0.23   -14.400 7.300   21.700  a:0.304 c:0.200 g:0.199 t:0.297
       3       Stat92E 9       4       1       1.5e-11  0.11   -12.600 10.700  23.300  a:0.304 c:0.200 g:0.199 t:0.297
       4       bcd     6       4       1       5.7e-12  0.23   -17.700 6.600   24.300  a:0.304 c:0.200 g:0.199 t:0.297
       5       eve     15      4       1       4.7e-24 0.00057 -33.000 13.900  46.900  a:0.304 c:0.200 g:0.199 t:0.297
       6       gt      10      4       1       2.7e-16 0.0095  -22.000 8.900   30.900  a:0.304 c:0.200 g:0.199 t:0.297
       7       hb      8       4       1       1.2e-16  0.11   -25.300 8.100   33.400  a:0.304 c:0.200 g:0.199 t:0.297
       8       kni     8       4       1       2.9e-11 0.0092  -12.500 6.300   18.800  a:0.304 c:0.200 g:0.199 t:0.297
       9       pan     8       4       1       2.1e-13 0.028   -18.300 7.300   25.600  a:0.304 c:0.200 g:0.199 t:0.297
       10      prd     8       4       1       1.5e-13 0.029   -18.400 8.300   26.700  a:0.304 c:0.200 g:0.199 t:0.297
       11      tin     6       4       1       8.2e-11  0.41   -15.000 7.200   22.200  a:0.304 c:0.200 g:0.199 t:0.297
       12      ttk     7       4       1       1.3e-11  0.12   -15.400 8.500   23.900  a:0.304 c:0.200 g:0.199 t:0.297
Number of sequences scanned     1
Sum of sequence lengths5500
N residues      0
Matches per matrix
       matrix  name    matches scored
       1       Kr              3       10986
       2       Med             0       10990
       3       Stat92E         2       10984
       4       bcd             0       10990
       5       eve             6       10972
       6       gt              3       10982
       7       hb              7       10986
       8       kni             2       10986
       9       pan             0       10986
       10      prd             2       10986
       11      tin             0       10990
       12      ttk             5       10988
               TOTAL           30      131826
```

Figure 7 | A matrix-scan result for the detection of individual sites. The header indicates the parameters used for the analysis. Predicted sites are displayed in a table, where each row corresponds to a predicted site, defined by its sequence, its coordinates on the input sequence and a series of scores. At the bottom of the table, the program returns some postscanning statistics (properties of the matrices, number of matches, running time).

▲ **CRITICAL STEP** The detection of CRERs to predict CRMs is a delicate issue, and the results will be drastically affected by the choice of the following parameters: the site *P*-value (**Box 2**), the background model (**Box 3**) and the CRER significance (**Box 4**). To obtain more reliable predictions, we suggest a two-step strategy. An overview of the CRER landscape is first obtained with permissive parameters (Steps 22 and 23); more stringent parameters are secondarily applied to filter the noise (Step 24).

22| Run the CRER detection with very permissive parameters to obtain a visual representation of the CRER landscape. Set the 'Upper threshold' on `site P-value` to 0.001 (note that with such a permissive threshold, we expect ~6 matches by chance for each one of the 12 PSSMs used in the study case when scanning on a single strand). Set the 'lower threshold' of `crer_sig` on 0 (note that this is also a very permissive threshold, we basically expect one false positive for every tested window). Click `GO`. After a few seconds, the result appears as a table where each row represents one CRER, characterized by its position and several score fields. Not surprisingly, this table is quite large because we deliberately over-predicted the CRERs.

23| Go to the bottom of the result table, click on the `feature-map` button, and execute Steps 17–19 to produce a feature map of the CRER landscape. It is recommended to inactivate the option `Legend`, because the legend will produce a huge list of CRER identifiers with the permissive parameters.

24| Run the CRER detection with more stringent parameters to predict a restricted number of CRERs that have a reasonable chance to correspond to CRMs. Come back to the matrix-scan form (as in Step 21) and set the `site P-value` to 0.0001 (under this threshold, we expect less than one false prediction per PSSM for the individual sites in the 5.5-kb region of the *eve* promoter). Set the 'Lower threshold' of `crer_sig` to 2 (above this threshold, we expect one false positive for every 100 tested CRERs). Click `GO`. After a few seconds, the results should be displayed. Generate a feature-map of the CRER predictions as in Step 23.

BOX 4 | PREDICTION OF *CIS*-REGULATORY ELEMENT ENRICHED REGIONS

The approach developed in *matrix-scan* to predict *cis*-regulatory modules is based on the detection of short regions that show a significant density of putative *cis*-regulatory elements. Such *cis*-regulatory element-enriched regions (CRERs) have a higher number of predicted site occurrences than expected by chance. First, individual sites are predicted using each matrix provided as input to the program. They are filtered using a threshold on their *P*-value (e.g., 0.001). Windows of variable sizes are then defined over the input sequence and the number of predicted sites (matches) is counted within each window. Second, the expected number of matches is calculated within each window and then serves to estimate the significance of the over-representation of matches. The binomial distribution has been proposed for the detection of over-represented words[41] and extended to the detection of over-represented position-specific scoring matrix matches[27]. The binomial distribution is used to estimate the probability to observe by chance at least as many matches as those counted in the window. This probability, called 'CRER *P*-value', has to be distinguished from the 'site *P*-value' (**Box 2**): the site *P*-value estimates the risk of false positive for an individual match, whereas the CRER *P*-value estimates the risk of error when considering that a window contains more matches than expected by chance. A threshold can be applied on the CRER *P*-value or CRER significance to return only those windows with a significant over-representation of matches.

Random controls

25| Click on the `Sequence retrieval` button in the left menu and select `random sequences`. The 'random-sequence' form allows choosing the number of desired sequences and their length. In the example, set the `Sequence length` to '5500' and `Number of sequences` to 10. In the `Nucleotide probabilities` section, select the background model that will serve to produce the random sequence. It can be the same background model as used for the scanning steps, or a higher order background model. In the example, select `Markov chain` and choose *Drosophila melanogaster* for the 'organism'. Keep `oligonucleotide size` to 6 to use a Markov background model of order 5 (see **Box 3** for details on Markov models). Click on the `GO` button to generate the sequences. The result page should contain the random sequence set, followed by the `Next step` section. Click on the `matrix-scan` button and repeat Steps 10–24 of the PROCEDURE with the same parameters as for the original input sequences.

! CAUTION In this study case, we use only ten sequences for the random control data set. It is however recommended to use a larger data set to better evaluate the random expectation.

▲ CRITICAL STEP To evaluate the quality of the predictions, it is crucial to rerun the analysis with sequences that serve as negative control. Such random controls may be run with either artificially generated sequences based on the background word frequencies (Step 25) or randomly picked biological sequences (Step 26).

26| Click on the `Genomes and genes` button in the left menu bar and select `random gene selection`. In the random gene selection form, specify the desired number of genes and the organism of interest. In the example, set `Number of genes` to ten and select *Drosophila melanogaster* for the `organism`. Click on the `GO` button. The result page displays the selected genes identifiers and the button `retrieve sequences` in the `Next step` section. Click on the button and repeat Steps 5–24 of the PROCEDURE with the same parameters as for the original input sequences.

TROUBLESHOOTING

Troubleshooting advice can be found in **Table 1**.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution
2	The genome of interest does not appear in the list of organisms	The genome of interest is not supported in regulatory sequence analysis tool	Sequences can be extracted from external sequence database such as University of California Santa Cruz Genome browser[38] (http://genome.ucsc.edu/). Go directly to Step 8 and load sequences in matrix-scan form
8	After a few minutes, the result of sequence retrieval is still not displayed	Due to the large size of the higher eukaryotes genomes, under some conditions, retrieving sequences may take a few minutes, particularly if there are many gene queries (e.g., retrieving all promoters of the human genome)	If you face this problem, restart the query by selecting the 'email' output in the sequence retrieval form instead of 'server'
9	In the sequence retrieval result, some sequences are missing and a message 'invalid query' appears	Some gene identifiers were not found in the genome annotations	Make sure the option `Query contains only IDs (no synonyms)` is not checked. Retry with a synonym (common name of the gene, gene identifier from another database). Alternatively, the `gene information tool` (in 'Genomes and genes' menu) can be used to find gene identifiers on the basis of a piece of their name or description

(continued)

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
16	The matrix-scan result does not appear after a long waiting time	For a large data set with computer-intensive options, the analysis can take several minutes, or even hours to run	Select the 'email' output in the matrix-scan form instead of 'display'. Alternatively, a more stringent threshold can be applied (e.g., discard negative scores with a lower threshold of 0 on the weight score)
16	There are too many matches to display and results are difficult to interpret	The threshold may be too loose and allows many false predictions. Another possibility is that your background model does not correspond to the composition of the sequence	Relaunch the analysis with a more stringent threshold. To obtain an idea about the number of matches expected by chance for a given weight threshold, you can use the program *matrix-distrib*

ANTICIPATED RESULTS

At the end of this protocol, the user should be able to visualize predicted sites and CRERs along the input sequence. In addition, several random controls should have been run to evaluate the quality of the predictions. **Figure 8** shows the feature-maps obtained with the *even-skipped* study case.

Figure 8a displays the binding sites (top) and CRMs (bottom) annotated in the 5,500-bp region located upstream the *even-skipped* TSS. TFBSs were extracted from ORegAnno[29] and CRMs from REDfly[31,37]. The right side of the figure corresponds to the region immediately upstream of the *even-skipped* gene whereas the left side of the figure corresponds to the most distal region. The coordinates of these elements are provided as supplementary material on the website (see *oreganno_eve_annotation.ft* for TFBSs and *redfly_eve_annotation.ft* for CRMs). **Figure 8a** can be reproduced by copy-pasting the content of these files in the *feature-map* form. The figure shows that the annotated TFBSs for the 12 transcription factors of interest fall into the four CRMs of the *even-skipped* promoter. The perfect correspondence between the annotated TFBSs and CRMs probably reflects some experimental or annotation bias and should by no means be taken as an evidence that these factors do not bind anywhere else in the region.

Figure 8b displays *matrix-scan* predictions in the *even-skipped* promoter. Individual site predictions are mostly found inside or in the neighborhood of the annotated CRMs. CRER predictions with the first set of parameters (Pval $\leq$ 0.001, crer_sig $\geq$ 0) show a landscape with many overlapping CRERs.

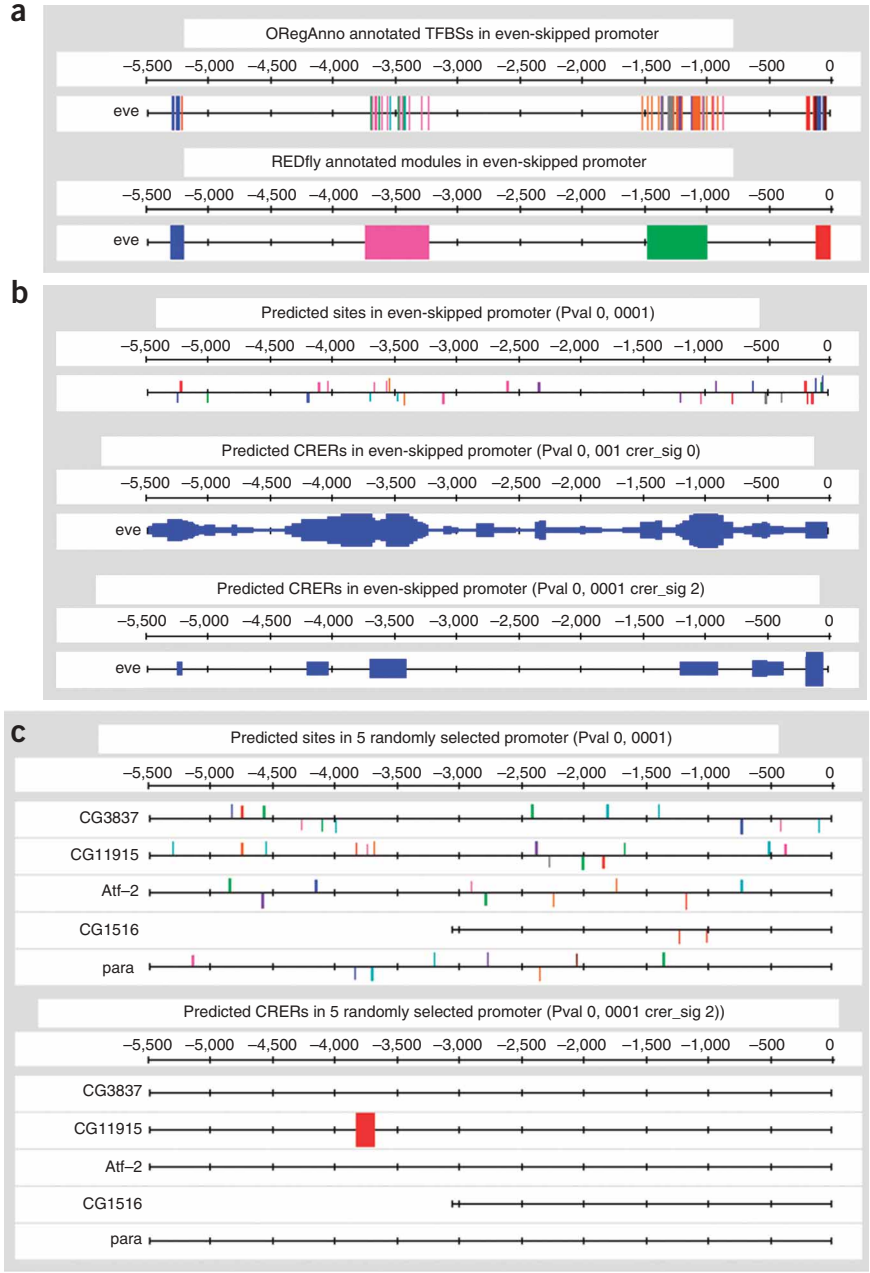

Figure 8 | Feature-maps for the *even-skipped* example. (**a**) Annotated transcription factor-binding sites (TFBSs) and *cis*-regulatory modules (CRMs) in the *even-skipped* promoter. (**b**) Matrix-scan predictions of sites and *cis*-regulatory element-enriched regions (CRERs) in the *even-skipped* promoter. (**c**) Matrix-scan predictions of sites and CRERs in randomly selected *drosophila* promoters.

Three regions seem to be of particular interest with highly significant CRERs (approximately −5200, −4000 and −1000). With the more stringent set of parameters (Pval ≤ 0.0001, crer_sig ≥ 2), noise is reduced and these regions appear more clearly. CRER predictions are arranged into six regions, four of which are localized in the annotated CRMs. *Matrix-scan* thus manages to detect the four annotated CRMs. Two additional CRERs are predicted around position −500 and −4000. These predictions might correspond to binding sites that are not yet annotated. Isolated individual sites predicted between positions −2000 and −3000 are not included in predicted CRERs and might correspond to false positive predictions. In this study case, CRERs comprise heterotypic and homotypic combinations of sites. To limit the study to homotypic CRERs, the user should submit only one matrix to the program.

Figure 8c illustrates *matrix-scan* results on five randomly picked promoter sequences. The fairly large number of individual site predictions in random sequences reveals the expected rate of false predictions. Note that it cannot be excluded that randomly picked biological sequences actually contain true binding sites. Prediction of CRERs nevertheless returns very few hits, which suggests that presence of a significant CRER is a good sign for putative binding sites. Predicted sites located in CRERs are thus more likely to be biologically relevant and are good candidates for experimental validation.

ACKNOWLEDGMENTS This work was supported by the Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture, FRIA (J.-V.T. PhD grant), the Vrije Universiteit Brussel (Geconcerteerde Onderzoeksactie 29) (M.T.-C. PhD grant), and by the BioSapiens Network of Excellence funded under the sixth Framework program of the European Communities (LSHG-CT-2003-503265). The postdoctoral grant of M.D. was funded by the Belgian Program on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, project P6/25 (BioMaGNet).

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

1. Wasserman, W.W. & Sandelin, A. Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet.* **5**, 276–287 (2004).
2. van Helden, J. Regulatory sequence analysis tools. *Nucleic Acids Res.* **31**, 3593–3596 (2003).
3. van Helden, J., André, B. & Collado-Vides, J. A web site for the computational analysis of yeast regulatory sequences. *Yeast* **16**, 177–187 (2000).
4. Defrance, M., Janky, R., Sand, O. & van Helden, J. Using RSAT oligo-analysis and dyad-analysis tools to discover regulatory signals in nucleic sequences. *Nat. Protoc.* doi:10.1038/nprot.2008.98 (2008).
5. Sand, O., Thomas-Chollier, M., Vervisch, E. & van Helden, J. Analyzing multiple data sets by interconnecting RSAT programs via SOAP Web services—an example with ChIP-chip data. *Nat. Protoc.* doi:10.1038/nprot.2008.99 (2008).
6. 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).
7. Wingender, E. TRANSFAC, TRANSPATH and CYTOMER as starting points for an ontology of regulatory networks. *In Silico Biol.* **4**, 55–61 (2004).
8. Wingender, E., Dietze, P., Karas, H. & Knüppel, R. TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Res.* **24**, 238–241 (1996).
9. Gama-Castro, S. *et al.* RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res.* **36**, D120–D124 (2008).
10. 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).
11. Hertz, G.Z. & Hartzell, G.W. 3rd & Stormo, G.D. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* **6**, 81–92 (1990).
12. Hertz, G.Z. & Stormo, G.D. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15**, 563–577 (1999).
13. Coessens, B. *et al.* INCLUSive: a web portal and service registry for microarray and regulatory sequence analysis. *Nucleic Acids Res.* **31**, 3468–3470 (2003).
14. Thijs, G. *et al.* A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17**, 1113–1122 (2001).
15. Kel, A.E. *et al.* MATCH: a tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.* **31**, 3576–3579 (2003).
16. Frith, M.C., Li, M.C. & Weng, Z. Cluster-Buster: finding dense clusters of motifs in DNA sequences. *Nucleic Acids Res.* **31**, 3666–3668 (2003).
17. Philippakis, A.A., He, F.S. & Bulyk, M.L. Modulefinder: a tool for computational discovery of *cis* regulatory modules. *Pac. Symp. Biocomput.* 519–530 (2005).
18. Sosinsky, A., Bonin, C.P., Mann, R.S. & Honig, B. Target Explorer: an automated tool for the identification of new target genes for a specified set of transcription factors. *Nucleic Acids Res.* **31**, 3589–3592 (2003).
19. Donaldson, I.J., Chapman, M. & Göttgens, B. TFBScluster: a resource for the characterization of transcriptional regulatory networks. *Bioinformatics* **21**, 3058–3059 (2005).
20. Donaldson, I.J. & Göttgens, B. TFBScluster web server for the identification of mammalian composite regulatory elements. *Nucleic Acids Res.* **34**, W524–W528 (2006).
21. Berman, B.P. *et al.* Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in *Drosophila melanogaster* and *Drosophila pseudoobscura*. *Genome Biol.* **5**, R61 (2004).
22. Berman, B.P. *et al.* Exploiting transcription factor binding site clustering to identify *cis*-regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc. Natl. Acad. Sci. USA* **99**, 757–762 (2002).
23. Pierstorff, N., Bergman, C.M. & Wiehe, T. Identifying *cis*-regulatory modules by combining comparative and compositional analysis of DNA. *Bioinformatics* **22**, 2858–2864 (2006).
24. Aerts, S., Van Loo, P., Moreau, Y. & De Moor, B. A genetic algorithm for the detection of new *cis*-regulatory modules in sets of coregulated genes. *Bioinformatics* **20**, 1974–1976 (2004).
25. Loots, G.G. & Ovcharenko, I. rVISTA 2.0: evolutionary analysis of transcription factor binding sites. *Nucleic Acids Res.* **32**, W217–W221 (2004).
26. Levine, M. & Tjian, R. Transcription regulation and animal diversity. *Nature* **424**, 147–151 (2003).
27. Aerts, S. *et al.* Toucan: deciphering the *cis*-regulatory logic of coregulated genes. *Nucleic Acids Res.* **31**, 1753–1764 (2003).
28. Stanojevic, D., Small, S. & Levine, M. Regulation of a segmentation stripe by overlapping activators and repressors in the *Drosophila* embryo. *Science* **254**, 1385–1387 (1991).
29. Montgomery, S.B. *et al.* ORegAnno: an open access database and curation system for literature-derived promoters, transcription factor binding sites and regulatory variation. *Bioinformatics* **22**, 637–640 (2006).
30. Griffith, O.L. *et al.* ORegAnno: an open-access community-driven resource for regulatory annotation. *Nucleic Acids Res.* **36**, D107–D113 (2008).
31. Halfon, M.S., Gallo, S.M. & Bergman, C.M. REDfly 2.0: an integrated database of *cis*-regulatory modules and transcription factor binding sites in *Drosophila*. *Nucleic Acids Res.* **36**, D594–598 (2008).
32. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. & Wheeler, D.L. GenBank. *Nucleic Acids Res.* **35**, D21–D25 (2007).
33. Flicek, P. *et al.* Ensembl 2008. *Nucleic Acids Res.* **36**, D707–D714 (2008).
34. Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W. & Lenhard, B. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* **32**, D91–D94 (2004).
35. Vlieghe, D. *et al.* A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Res.* **34**, D95–D97 (2006).
36. 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).
37. Gallo, S.M., Li, L., Hu, Z. & Halfon, M.S. REDfly: a regulatory element database for *Drosophila*. *Bioinformatics* **22**, 381–383 (2006).
38. Bina, M. The genome browser at UCSC for locating Genes, and much more! *Mol. Biotechnol.* **38**, 269–275 (2008).
39. Staden, R. Methods for calculating the probabilities of finding patterns in sequences. *Comput. Appl. Biosci.* **5**, 89–96 (1989).
40. Robin, S., Rodolphe, F. & Schbath, S. *DNA, Words and Models—Statistics of Exceptional Words* (Cambridge University Press, Cambridge, U.K., 2005).
41. 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).