Regulatory Sequence Analysis Tools

Jacques van Helden*

Service de Conformation des Macromolécules Biologiques et de Bioinformatique, Université Libre de Bruxelles, Campus Plaine, CP 263, Bld du Triomphe, B-1050 Bruxelles, Belgium

Received February 14, 2003; Revised and Accepted April 3, 2003

ABSTRACT
The web resource Regulatory Sequence Analysis Tools (RSAT) (http://rsat.ulb.ac.be/rsat) offers a collection of software tools dedicated to the prediction of regulatory sites in non-coding DNA sequences. These tools include sequence retrieval, pattern discovery, pattern matching, genome-scale pattern matching, feature-map drawing, random sequence generation and other utilities. Alternative formats are supported for the representation of regulatory motifs (strings or position-specific scoring matrices) and several algorithms are proposed for pattern discovery. RSAT currently holds >100 fully sequenced genomes and these data are regularly updated from GenBank.

INTRODUCTION
Despite the essential role played by non-coding sequences in transcriptional regulation, genome annotations usually focus on identifying the genes and predicting their function through sequence similarity searches. The services offered by most genome centers are restricted to analysis of coding and peptidic sequences. The web resource Regulatory Sequence Analysis Tools (RSAT) is dedicated to the analysis of the other part of the genomes: the non-coding sequences. It proposes a collection of modular tools which can be combined in different ways to predict regulatory elements. Three main scenarios can be handled: (i) starting from a set of co-regulated genes, retrieve their upstream sequences and detect over-represented motifs, which might be responsible for their co-regulation (pattern discovery); (ii) predict the location of binding sites for a known transcription factor in a given sequence (pattern matching); (iii) starting from the known consensus pattern for a given transcription factor, scan all upstream sequences of a selected genome in order to predict putative target genes (genome-scale pattern matching).

TASKS AND PROGRAMS
The procedures currently supported by RSAT are summarized in Table 1. These procedures are linked in a pipeline as illustrated in Figure 1. In the following we describe different tasks and the programs that are most appropriate for performing them.

Sequence retrieval
The simplest input for RSAT is a list of gene names. Using this list the retrieve-seq program returns upstream, downstream or unspliced ORF sequences (introns and spliced ORFs will soon be supported). The user can specify the left and right limits of the sequences to be retrieved. Default values have been selected for each genome, depending on the average size of the intergenic regions and mechanisms of regulation. Upstream sequences can be retrieved over a constant size, but an option also allows to clip them in order to avoid the inclusion of coding sequences from upstream ORFs.

Transcription factor binding sites
The specificity of a transcription factor can be described by a pattern. Two alternative formats are currently used to describe regulatory signals: strings (including the IUPAC alphabet for ambiguous nucleotides) or position-specific scoring matrices (PSSM) (1).

Pattern matching
When the regulatory pattern is known (e.g. the consensus binding sequence for a given transcription factor), one may wish to locate its occurrences, in order to identify putative transcription factor binding sites in upstream sequences of a set of genes. Patterns can be collected from the literature or obtained from specialized databases (2–4). String-based pattern matching is performed with the program dna-pattern. This program supports the IUPAC degenerate alphabet, as well as regular expressions, which allow the specification of spaces of variable length. Patterns can be searched on either one or both strands. A matrix-based pattern matching procedure, patser, developed by Jerry Hertz (5,6), has been integrated to the web interface.

Pattern discovery
Given a set of co-regulated genes, pattern discovery programs can be used to detect over-represented motifs in their upstream regions. This is particularly useful for the prediction of regulatory motifs from clusters of co-expressed genes, such as
those obtained from microarray data or other high-throughput methods. Several algorithms for pattern discovery are supported. The program *oligo-analysis* (7) analyzes oligonucleotide occurrences and returns those that are statistically over-represented (Table 2A).

Despite its simplicity, this program has proven to be very efficient for the detection of regulatory motifs in the yeast *Saccharomyces cerevisiae*. However, some motifs escape detection, because they take the form of a spaced dyad, i.e. a pair of very short oligonucleotides separated by a region of fixed length but variable content. A second program, *dyad-analysis* (8), specifically detects such spaced motifs, which are typical of many bacterial transcription factors, and of the fungal binuclear zinc cluster proteins. String-based pattern
discovery programs generally return several oligonucleotides or dyads, which can be assembled with the program pattern-assembly, to yield larger and/or partially degenerate motifs (Table 2B). Two matrix-based pattern discovery programs, Andrew Neuwald’s gibbs sampler (9) and Jerry Hertz’s consensus (5,6), are also available.

The strength of string-based pattern discovery methods is their very low rate of false positives and the fact that they are able to return multiple motifs when a set of genes is regulated by several factors. This is illustrated by the example in Table 2B, where the analysis of 10 methionine-responsive genes led to the detection of two distinct patterns, corresponding to the binding sites of Met4p and Met31p, respectively. Matrix-based programs return a more refined description of pattern degeneracy, but have the drawback of always returning an answer, even when random sequences are submitted.

Genome-scale pattern matching

Pattern matching can be applied to the full set of upstream sequences in a genome, in order to predict genes possibly regulated by a given transcription factor. It should be noted that the simple presence of a motif in a given upstream region is generally not sufficient to predict regulation. Indeed, given the short size of the motifs and the large size of the genomes, hundreds, or even thousands of matches could be returned by chance alone. Predictions can be improved by detecting multiple binding sites, either for the same transcription factor, or for combinations of several different transcription factors.

Feature map drawing

The results obtained by pattern matching can be displayed graphically, in the form of a feature map (Fig. 2). In this map, each motif is represented by a box painted in a different color, whose height is proportional to the statistical significance of the pattern. Feature maps are not only useful for illustrative purposes, they can also reflect additional properties of the discovered motifs such as a conserved position relative to the start codon, a distal or proximal location, the pairing of heterologous motifs and so on.

Random sequences and random gene selections

Random sequences are useful for performing negative controls. Indeed, some programs present the inconvenience of systematically returning an answer, even when the submitted sequence set contains no biologically significant features. The program random-seq generates random DNA sequences on the basis of the selected background model (e.g. intergenic sequences from S. cerevisiae). For each pattern, the number of occurrences (occ) is compared to the random expectation (exp_occ), and the P-value is calculated (occ_P). The significance index (occ_sig) equals -log (P-value). Over-represented patterns have positive occ_sig values. In the sequences analyzed here, no more than seven patterns were significant among the 2080 possibilities. Notice that some of the patterns strongly overlap with each other.

(B) The seven patterns from Table A were aligned with the program pattern-assembly, resulting in two clusters of hexanucleotides, each forming a larger consensus. The most significant result, TCACGTGA, is the consensus of the Met4p/Cbf1p/Met28p complex. The second consensus, AACTGTGGC, corresponds to the binding site for Met31p.

Upstream sequences were retrieved >800 bp from the start codon. (A) Over-represented oligonucleotides. Each row represents one pattern, i.e. a hexanucleotide with its reverse complement. Each pattern has a specific prior probability (exp_freq) defined on the basis of the selected background model (e.g. intergenic sequences from S. cerevisiae). For each pattern, the number of occurrences (occ) is compared to the random expectation (exp_occ), and the P-value is calculated (occ_P). The significance index (occ_sig) equals -log (P-value). Over-represented patterns have positive occ_sig values. In the sequences analyzed here, no more than seven patterns were significant among the 2080 possibilities. Notice that some of the patterns strongly overlap with each other.

(B) The seven patterns from Table A were aligned with the program pattern-assembly, resulting in two clusters of hexanucleotides, each forming a larger consensus. The most significant result, TCACGTGA, is the consensus of the Met4p/Cbf1p/Met28p complex. The second consensus, AACTGTGGC, corresponds to the binding site for Met31p.

**Table 2.** Output of oligo-analysis applied to a set of 12 yeast genes regulated by methionine (SAM2, MET6, MUP3, MET30, MET3, MET14, MET1, SAM1, MET17, ZWF1 and MET2)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>exp_freq</th>
<th>occ</th>
<th>exp_occ</th>
<th>occ_P</th>
<th>occ_E</th>
<th>occ_sig</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACGCTT</td>
<td>0.000174</td>
<td>15</td>
<td>0.000559</td>
<td>2.06</td>
<td>0.05</td>
<td>0.001</td>
<td>1</td>
</tr>
<tr>
<td>ACAGCT</td>
<td>0.000559</td>
<td>11</td>
<td>0.002118</td>
<td>0.56</td>
<td>0.02</td>
<td>0.01</td>
<td>2</td>
</tr>
<tr>
<td>ACTGTA</td>
<td>0.000559</td>
<td>14</td>
<td>0.002118</td>
<td>0.56</td>
<td>0.02</td>
<td>0.01</td>
<td>3</td>
</tr>
<tr>
<td>ACTGTA</td>
<td>0.000559</td>
<td>14</td>
<td>0.002118</td>
<td>0.56</td>
<td>0.02</td>
<td>0.01</td>
<td>3</td>
</tr>
<tr>
<td>ACACCA</td>
<td>0.000295</td>
<td>10</td>
<td>0.000058</td>
<td>4.06</td>
<td>0.001</td>
<td>0.01</td>
<td>4</td>
</tr>
<tr>
<td>ACGCTT</td>
<td>0.000413</td>
<td>12</td>
<td>0.000116</td>
<td>0.42</td>
<td>0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
</tbody>
</table>

**Figure 1.** Flow chart of the Regulatory Sequence Analysis Tools.
circumstances. The approaches used have strong points as well as limitations of which one should be well aware. Any predictive method will unavoidably return false positives and/ or miss some genuine regulatory patterns. Until now, most methods have been optimized and validated in microbial model organisms (yeast and bacteria). Whether these approaches can be extended to higher organisms is still an open question. Currently we are evaluating the applicability of RSAT for the detection of regulatory elements in the genomes of multi-cellular organisms. To perform such detection successfully, new methods, based on comparative genomics, might be required (reviewed in 10).

ACKNOWLEDGEMENTS

I acknowledge support from the Action de Recherches Concertées de la Communauté Française de Belgique (contract ARC-97/01-211). I am grateful to Jean Richelle for the system administration and to Shoshana Wodak who helped to improve this manuscript. The RSAT project was originated at the Universidad Nacional Autonoma de Mexico, in the laboratory of Julio Collado-Vides, to whom I am thankful for past and present collaboration.

REFERENCES