BIOINFORMATICS APPLICATIONS NOTE Vol. 18 no. 8 2002
Pages 1135–1136

TFBS: Computational framework for transcription factor binding site analysis

Boris Lenhard and Wyeth W. Wasserman*

Bioinformatics Unit, Center for Genomics and Bioinformatics, Karolinska Institutet, Stockholm, Sweden

Received on December 20, 2001; revised on February 22, 2002; accepted on February 28, 2002

ABSTRACT

Summary: TFBS is a set of integrated, object-oriented Perl modules for transcription factor binding site detection and analysis. It implements objects representing specificity profile matrices, binding sites and sets thereof, pattern generators, and pattern database interfaces. The modules are interoperable with the BioPerl open source system.

Availability and Supplementary Information: The module package with documentation and example scripts are available at http://forkhead.cgb.ki.se/TFBS/.

Contact: Boris.Lenhard@cgb.ki.se

INTRODUCTION

Eukaryotic regulatory regions are characterized by the presence of multiple transcription factor binding sites, which can be described as sequence patterns with varying degrees of degeneracy. For computational analysis of regulatory regions, most approaches can be described by three sequential phases. First, a pattern is described for the set of target sequences known to be bound by a specific transcription factor. Second, a set of DNA sequences are analysed to determine the location(s) of sequences consistent with the described binding pattern (Staden, 1990). Finally, in advanced cases, predictive statistical models of regulatory regions are constructed based on multiple occurrences of the detected patterns (Wasserman and Fickett, 1998).

Frequency matrices have proven to be the most successful quantitative method for representing the binding specificity of a given transcription factor (Berg and von Hippel, 1988). From this foundational unit, a number of quantitative variations have been successfully applied, including weight matrices for easier statistical analysis and information content matrices well-suited for graphical representations (Schneider and Stephens, 1990) of the binding pattern. An annotated scheme for recording specificity of a sequence in a matrix and its transformation to other types of matrices is depicted in a figure at http://forkhead.cgb.ki.se/TFBS/matrixfigure.html.

*To whom correspondence should be addressed.

In regulatory regions, multiple transcription factor binding sites are frequently found clustered in short stretches of 50–200 nucleotides, such clusters are often called *regulatory modules* (Arnone and Davidson, 1997). This fact has been exploited for building predictive models for tissue-specific regulatory regions in mammals (Wasserman and Fickett, 1998; Krivan and Wasserman, 2001), as well as more stringent rule-based models (Frech *et al.*, 1997).

Recent progress has been made in addressing two 'time' challenges in bioinformatics: incompatible data formats and frequently encountered re-programming tasks (see e.g. http://www.bioperl.org, http://www.biopython.org). Yet, currently there are no standard formats for storage and exchange of patterns used for recognition of transcription factor binding sites, and many analyses are performed with idiosyncratic approaches that often fail to capitalize on past developments ('reinventing the wheel'). Here we present TFBS, a set of object-oriented Perl modules for transcription factor detection and analysis, which seamlessly integrates generation, manipulation, storage and retrieval of patterns for transcription factor binding sites, as well as scanning sequences and alignments of sequences for matches to these patterns. It reduces program coding time, enabling computational biologists to explore biologically meaningful topics rather than managing low-level data structures.

OVERVIEW AND EXAMPLES

The classes constituting TFBS encompass the storage, manipulation and analysis objects for nucleotide sequence patterns and the corresponding nucleotide sequence features. Currently available classes are listed in Table 1. Among these are *pattern classes, pattern set classes* and *pattern generators. Pattern classes* hold the actual matrix profiles with the associated information. *Pattern set classes* are aggregate classes for storage and manipulation of multiple matrices (e.g. for scanning nucleotide sequences with multiple matrices, when one wants to analyse the resulting set as a whole). *Pattern generators* are factory classes for constructing new matrices from a set of

Table 1. Currently available public classes in TFBS (internal, abstract and base classes are omitted)

Class	Description
TFBS::Matrix::PFM	Position frequency (raw count) matrix; can be transformed to other matrix types
TFBS::Matrix::ICM	Information content matrix; used e.g. for drawing sequence logos
TFBS::Matrix::PWM	Position weight matrix; used for scanning DNA sequences and alignments thereof (phylogenetic footprinting)
TFBS::Site	A sequence feature object for transcription factor binding site
TFBS::SiteSet	An aggregate of TFBS::Site objects, with methods for manipulation of the set
TFBS::SitePair	A sequence feature object for pair of transcription factor binding sites in orthologous sequences
TFBS::SitePairSet	An aggregate of TFBS::SitePair objects, with methods for manipulation of the set
TFBS::DB::FlatFileDir	Read/write interface to a simple, flat-file database of matrix object data
TFBS::DB::JASPAR2	Read/write interface to a MySQL database of matrix object data; data model described in the documentation
TFBS::DB::TRANSFAC	Read-only interface to public TRANSFAC database on the web
TFBS::PatternGen::SimplePFM	Pattern generator that creates a position weight matrix from a set of short sequences of fixed size
TFBS::PatternGen::Gibbs	Pattern generator that uses external Gibbs sampling program (Lawrence *et al.*, 1993) to find subtle patterns in a set of DNA sequences. For the program, write to C.E.Lawrence (see http://www.wadsworth.org/resnres/bioinfo/).

sequences using different algorithms, some of them implemented by existing external programs. For pattern classes, the basic functionality is shown in annotations to the figure at http://forkhead.cgb.ki.se/TFBS/matrixfigure.html.

The following two code snippets demonstrate the ease of use of TFBS objects:

- a script that retrieves a sequence from GenBank using BioPerl, a C/EBP position weight profile from TRANSFAC, scans the sequence with the matrix and outputs the detected sites in GFF format:

```
#!/usr/bin/env perl -w
use Bio::DB::GenBank;
use TFBS::DB::TRANSFAC;
my $seq = Bio::DB::GenBank->new()->
     get_Seq_by_acc('AF100993');
my $db = TFBS::DB::TRANSFAC->connect();
my $pwm = $db->get_Matrix_by_ID
     ('V$CEBPA_01','PWM');
my $siteset = $pwm->search_seq(-seqobj=>
     $seq, -threshold=>"80%");
print $siteset->GFF();
```

- a script that identifies new patterns from a set of DNA sequences stored in the file 'sequences.fa' and stores them in a simple flat-file database:

```
#!/usr/bin/env perl -w
use TFBS::DB::FlatFileDir;
use TFBS::PatternGen::Gibbs;
my $gibbs = TFBS::PatternGen::Gibbs->new
     (-file=>'sequences.fa',
-motif_length=>10);
my $db = TFBS::DB::FlatFileDir->create
     ('NewPatterns');
$db->store_Matrix($gibbs->
     all_patterns());
```

The annotated versions of the above and more complex example scripts are available at http://forkhead.cgb.ki.se/TFBS/.

FUTURE DEVELOPMENT

The TFBS functionality can be extended and expanded in a multitude of directions. Two additions currently under construction are:

- TFBS::DB:: SimpleXML : database interface for a flexible single-file exchange format for matrix patterns
- TFBS::Analysis::LRA::*—a set of classes for model building and statistical evaluation using logistic regression

Although originally designed for transcription factor patterns and binding sites, the overall design readily lends itself to the utilization of diverse patterns classes.

REFERENCES

Arnone,M.I. and Davidson,E.H. (1997) The hardwiring of development: organization and function of genomic regulatory systems. *Development*, **124**, 1851–1864.

Berg,O.G. and von Hippel,P.H. (1988) Selection of DNA binding sites by regulatory proteins. *Trends Biochem. Sci.*, **13**, 207–211.

Frech,K., Danescu-Mayer,J. and Werner,T. (1997) A novel method to develop highly specific models for regulatory units detects a new LTR in GenBank which contains a functional promoter. *J. Mol. Biol.*, **270**, 674–687.

Krivan,W. and Wasserman,W.W. (2001) A predictive model for regulatory sequences directing liver-specific transcription. *Genome Res.*, **11**, 1559–1566.

Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **262**, 208–214.

Schneider,T.D. and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.*, **18**, 6097–6100.

Staden,R. (1990) Searching for patterns in protein and nucleic acid sequences. *Methods Enzymol.*, **183**, 193–211.

Wasserman,W.W. and Fickett,J.W. (1998) Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.*, **278**, 167–181.