BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 8 2007, pages 1029–1031
doi:10.1093/bioinformatics/btm041

Sequence analysis

SPACER: identification of *cis*-regulatory elements with non-contiguous critical residues

Arijit Chakravarty[1], Jonathan M. Carlson[2], Radhika S. Khetani[3], Charles E. DeZiel[3] and Robert H. Gross[3,*]

[1]Department of Cancer Pharmacology, Millennium Pharmaceuticals Inc., Cambridge, MA, [2]Department of Computer Science and Engineering, University of Washington, Seattle, WA and [3]Department of Biology, Dartmouth College, Hanover, NH, USA

Received on November 12, 2006; revised on January 31, 2007; accepted on February 1, 2007

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Many transcription factors bind to sites that are long and loosely related to each other. *De novo* identification of such motifs is computationally challenging. In this article, we propose a novel semi-greedy algorithm over the space of all IUPAC degenerate strings to identify the most over-represented highly degenerate motifs.

Results: We present an implementation of this algorithm, named SPACER (Separated Pattern-based Algorithm for *cis*-Element Recognition) and demonstrate its effectiveness in identifying 'gapped' and highly degenerate motifs. We compare SPACER's performance against ten motif finders on 42 experimentally defined regulons from *Bacillus subtilis, Escherichia coli* and *Saccharomyces cerevisiae*. These motif finders cover a wide range of both enumerative and statistical approaches, including programs specifically designed for prokaryotic and 'gapped' motifs.

Availability: A Java 1.4 implementation is freely available on the Web at http://genie.Dartmouth.edu/SPACER/

Contact: robert.h.gross@dartmouth.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

The regulation of gene transcription is mediated via the alteration of promoter activity by DNA-binding proteins called transcription factors. The variability between the sequences of individual binding sites for the same transcription factor is position-specific, as certain bases are constrained by virtue of their contact with the transcription factor, while others are free to accumulate neutral mutations (Moses *et al*., 2003). Since selection acts to remove spurious transcription factor binding sites in the genome (Hahn *et al*., 2003), a set of binding sites for the same transcription factor (referred to collectively as a *cis*-regulatory element) should be computationally identifiable at the level of sequence alone, as a set of over-represented sequences contained within the upstream regions of interest.

Unfortunately, this simple intuition often breaks down in practice. If the bases that mediate protein–DNA interaction are non-contiguous, it is possible that a *cis*-regulatory element might not contain any over-represented subsequences. One family of transcription factors with widely spaced critical binding residues is the fungal Zn(II)2Cys6 binuclear cluster family, whose members include the *Saccharomyces cerevisiae* GAL4 protein (Pan and Coleman, 1989). The computational identification of Zn(II)2Cys6 family *cis*-regulatory elements is particularly difficult, because the sites are long and contain constrained bases only at their ends (for example, the consensus for the GAL4 *cis*-regulatory element is CCGnnnnnnnnnnnCGG). In more general terms, many transcription-factor-binding sites from other organisms are also very long and may contain one or more highly variable sections separated by critical residues that are less variable. For example, *cis*-regulatory elements in bacterial genomes are usually long (~30 bases) and highly variable. In the case of some bacterial *cis*-regulatory elements, most of the sequence signal is carried in two sub-regions, each ~6 bp in length (Robison *et al*., 1998).

2 RELATED WORK

Although over forty different algorithms exist for *de novo* motif identification (see Bulyk, 2003; MacIsaac and Fraenkel, 2006; Wasserman and Sandelin, 2004 and for reviews), many motif finders do not take into account the structure of gapped motifs and therefore perform poorly on them.

This finding has motivated the creation of a number of algorithms specifically aimed at the identification of gapped motifs. These algorithms include Bioprospector, BIPAD and SeSiMCMC (Bi and Rogan, 2004; Favorov *et al*., 2005; Liu *et al*., 2001), all based on a position weight matrix (PWM) motif model, and the program MITRA (Eskin and Pevzner, 2002), which is based on a k-mismatch motif model (in which a motif is represented as a word of length L with at most k mismatches). The programs RSAT (dyad-analysis) and YMF, both of which exhaustively enumerate motifs (using a consensus motif model), are also capable of specifically

*To whom correspondence should be addressed.

searching for gapped motifs (Sinha and Tompa, 2002; van Helden *et al*., 2000), although neither of these programs are able to identify weak base preferences in the spacer region, motifs of arbitrary length, or full degeneracies in the binding regions at the ends (both programs can search over a limited set of degeneracies).

In previous work, we have described two semi-greedy algorithms, BEAM and PRISM (Carlson *et al*., 2006a, b). BEAM is aimed at the identification of non-degenerate motifs and is based on the intuition that over-represented motifs will contain within them over-represented sub-motifs. PRISM is aimed at the identification of degenerate motifs with contiguous critical residues, based on the observation that over-represented degenerate motifs are comprised of over-represented non-degenerate motifs. By generalizing the output of BEAM, PRISM is able to identify motifs of arbitrary length and degree of degeneracy over the entire IUPAC alphabet (a 15-letter code consisting of the bases A,T,C,G and all resulting combinations). In performance comparisons using a large experimentally determined dataset, PRISM, with its focused search strategy, outperformed a number of other publicly available motif finders. (These motif finders were selected to represent a diversity of approaches, including expectation maximization, Gibbs sampling, exhaustive enumeration and heuristic mismatch algorithms.)

Although both BEAM and PRISM perform well on the class of motifs that they are designed to find, neither is expected to perform well in the detection of long, highly degenerate motifs with non-contiguous critical residues. BEAM, which works by extending short over-represented motifs to return motifs of any length, is unable to extend its motifs through the spacer region. PRISM relies on BEAM to identify seed motifs for further optimization. This approach is reasonable for over-represented motifs of low or moderate degeneracy, but breaks down for long, highly degenerate motifs (Carlson *et al*., 2006b).

3 RESULTS AND DISCUSSION

In this article, we present a program for the identification of long, highly variable motifs that contain one or more subregions of low sequence specificity. Our program, SPACER (Separated Pattern-based Algorithm for *cis*-Element Recognition), uses as its objective function the statistical over-representation of the motifs in the group of target upstream sequences, relative to all upstream regions in a given genome. SPACER's first stage, cSPACER (canonical SPACER), uses a beam search algorithm. Beam search algorithms, common in the field of natural language processing, are semi-greedy algorithms that simultaneously consider several promising paths in parallel, instead of focusing only on the most promising path (Russell and Norvig, 1995).

cSPACER identifies arbitrary length bipartite motifs of the form A-S_N-B, where A and B are independent and separated by a degenerate spacer region S_N (a string of n's). For each canonical bipartite motif, the non-degenerate ends (A and B) are systematically modified by PRISM to identify degeneracies that result in a greater degree of over-representation for the motif. In the final stage, the degenerate spacer region (S_N) is specialized to identify weak base preferences that result in a greater degree of over-representation of the given motif using the entire IUPAC alphabet (Supplementary Fig. 1). The final output of SPACER is a position weight matrix given by the individual sequences that correspond to the most over-represented degenerate motif.

Briefly, SPACER differs from PRISM and BEAM in the following respects:

- A different type of seed motif (gapped motifs instead of short contiguous sequences).
- A modification to the beam search algorithm, enabling motif extension to be performed in a greedy manner.
- A specialized suffix array, enabling rapid lookup for gapped and highly degenerate motifs.
- A novel algorithm that identifies weak base preferences in the spacer region.

A detailed description of the modifications of BEAM and PRISM that resulted in the creation of SPACER may be found in the Supplementary Materials.

We tested the full SPACER algorithm on a biological dataset of forty-two regulons taken from *S.cerevisiae*, *Escherichia coli* and *Bacillus subtilis* (Supplement Section 2.6). For *E.coli* and *B.subtilis*, we selected every available regulon containing at least five upstream sequences from the PRODORIC database. For *S.cerevisiae*, we selected every available regulon known to belong to transcription factors in the Zn(II)2Cys6 family. Quantitative measures of performance were established by comparing the motifs returned by SPACER against the known binding sites. The metrics employed were *accuracy* (the proportion of overlap between the published and predicted nucleotides; for details, see Supplement Section 2.7), *specificity* (fraction of predicted nucleotides that overlap with published) and *sensitivity* (fraction of published nucleotides that overlap with predicted). On this dataset, SPACER returned an average accuracy of 0.24, with a specificity of 0.41 and sensitivity of 0.35. Sequence logos generated by SPACER showed a reasonable match with published logos (Supplementary Fig. 2; for details see Supplement Section 3.1).

Next, we sought to place SPACER's performance on this dataset in context by comparing its results to those generated by BEAM and PRISM. On this dataset, SPACER outperformed these algorithms by a large margin. SPACER's average accuracy was 0.24, which was substantially higher than that of BEAM (0.13) and PRISM (0.14). This margin was statistically significant ($P<0.001$ for both comparisons against SPACER by a two-tailed paired *t*-test). Following Sinha and Tompa (2002), we looked at the frequency of clear wins in head-to-head comparisons, defined as an instance where one program outperformed the other program by a margin of at least 0.10. In all head-to-head comparisons against BEAM and PRISM, SPACER had the higher score in 84% of cases where a clear win was recorded (Supplementary Fig. 3).

To provide further context for SPACER's performance, we compared it to ten other popular motif-finding programs on this dataset, including several specifically aimed at the identification of bipartite motifs (Supplement Section 3.2).

Using the criteria originally proposed by Sinha and Tompa (2002), SPACER outperformed the other programs on this dataset (Supplementary Table 1). SPACER had the highest average accuracy, and the largest number of regulons with accuracy scores $\geq$0.50, $\geq$0.33 and $\geq$0.10. In addition, SPACER had more wins (12) than any other program (MEME and RSAT are the next highest, with 6). In head-to-head comparisons against all ten other programs where a clear win was recorded, SPACER had the higher score in 78% of the cases. SPACER scores among the highest accuracy for each of the three species in this dataset (Supplementary Fig. 4). Broken down by sensitivity and specificity criteria, SPACER performs strongly relative to the other programs (Supplementary Fig. 5), as the most specific and second-most sensitive (after AlignACE).

Finally, to assess the performance of the different programs under noisy conditions, we tested SPACER's performance on Zn(II)2Cys6 regulons (sets of genes regulated by the same transcription factor) mixed with randomly selected genes from the *S.cerevisiae* genome. In each regulon, we introduced between 1 and 4 times as many randomly selected genes as the number of genes originally present. Although the performance of the other programs tested deteriorated in the presence of noise, SPACER's performance on this dataset was superior at all levels of noise (Supplementary Fig. 6; Supplement Section 3.3). Such a proportional decrease in accuracy across the different programs resulted in a greater number of situations where SPACER's accuracy was higher than that of the other programs (93% of clear head-to-head wins).

4 CONCLUSION

In conclusion, SPACER represents a novel approach to the identification of bipartite *cis*-regulatory elements. SPACER is able to identify *cis*-regulatory elements of the form A-S_N-B where A and B are sequences of arbitrary length and degeneracy and S_N may contain some specificity, providing a degree of flexibility not typically achievable using consensus-based algorithms like YMF, for which enumeration of patterns longer than 3-n-3 becomes prohibitively expensive. In addition, the beam search algorithm employed by SPACER enables it to scale independently of the number of upstream sequences in the regulon. The beam search strategies that SPACER employs allow it to explore the parameter space without eliciting user estimates for any variables. On a dataset enriched for bipartite and long degenerate motifs, we found that SPACER's focused search over the entire IUPAC alphabet consistently outperformed PWM and mismatch algorithms, including those specifically aimed at the identification of bipartite motifs.

ACKNOWLEDGEMENTS

The authors would like to thank Nelson Rosa Jr for his help in automating test runs on the other programs. This research was supported by a grant to R.H.G. from the National Science Foundation, DBI-0445967.

Conflict of Interest: none declared.

REFERENCES

Bi,C. and Rogan,P.K. (2004) Bipartite pattern discovery by entropy minimization-based multiple local alignments. *Nucleic Acids Res.*, **32**, 4979–4991.

Bulyk,M.L. (2003) Computational prediction of transcription-factor binding site locations. *Genome Biol.*, **5**, 201.

Carlson,J.M. *et al.* (2006a) BEAM: A beam search algorithm for the identification of *cis*-regulatory elements in groups of genes. *J. Comput. Biol.*, **13**, 686–701.

Carlson,J.M. *et al.* (2006b) Bounded search for *de novo* identification of degenerate *cis*-regulatory elements. *BMC Bioinformatics*, **7**, 254.

Eskin,E. and Pevzner,P.A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, **18** (Suppl 1), S354–S363.

Favorov,A.V. *et al.* (2005) A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics*, **21**, 2240–2245.

Hahn,M.W. *et al.* (2003) The effects of selection against spurious transcription factor binding sites. *Mol. Biol. Evol.*, **20**, 901–906.

Liu,X. *et al.* (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, **6**, 127–138.

MacIsaac,K.D. and Fraenkel,E. (2006) Practical strategies for discovering regulatory DNA sequence motifs. *PLoS Comput. Biol.*, **2**, e36.

Moses,A.M. *et al.* (2003) Position specific variation in the rate of evolution in transcription factor binding sites. *BMC Evol. Biol.*, **3**, 19.

Pan,T. and Coleman,J.E. (1989) Structure and function of the Zn(II) binding site within the DNA-binding domain of the GAL4 transcription factor. *Proc. Natl Acad. Sci. USA*, **86**, 3145–3149.

Robison,K. *et al.* (1998) A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete *Escherichia coli* K-12 genome. *J. Mol. Biol.*, **284**, 241–254.

Russell,S. and Norvig,P. (1995) *Artificial Intelligence: A Modern Approach*. Prentice Hall, NJ, pp. 94–137.

Sinha,S. and Tompa,M. (2002) Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.*, **30**, 5549–5560.

van Helden,J. *et al.* (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.*, **28**, 1808–1818.

Wasserman,W.W. and Sandelin,A. (2004) Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet.*, **5**, 276–287.