W412–W416 *Nucleic Acids Research, 2005, Vol. 33, Web Server issue*
doi:10.1093/nar/gki492

WordSpy: identifying transcription factor binding motifs by building a dictionary and learning a grammar

Guandong Wang[1], Taotao Yu[1] and Weixiong Zhang[1,2,*]

[1]Department of Computer Science and Engineering and [2]Department of Genetics, Washington University in Saint Louis, Saint Louis, MO 63130, USA

Received February 14, 2005; Revised and Accepted April 25, 2005

ABSTRACT

Transcription factor (TF) binding sites or motifs (TFBMs) are functional *cis*-regulatory DNA sequences that play an essential role in gene transcriptional regulation. Although many experimental and computational methods have been developed, finding TFBMs remains a challenging problem. We propose and develop a novel dictionary based motif finding algorithm, which we call WordSpy. One significant feature of WordSpy is the combination of a word counting method and a statistical model which consists of a dictionary of motifs and a grammar specifying their usage. The algorithm is suitable for genome-wide motif finding; it is capable of discovering hundreds of motifs from a large set of promoters in a single run. We further enhance WordSpy by applying gene expression information to separate true TFBMs from spurious ones, and by incorporating negative sequences to identify discriminative motifs. In addition, we also use randomly selected promoters from the genome to evaluate the significance of the discovered motifs. The output from WordSpy consists of an ordered list of putative motifs and a set of regulatory sequences with motif binding sites highlighted. The web server of WordSpy is available at http://cic.cs.wustl.edu/wordspy.

INTRODUCTION

Knowledge of gene regulation is of fundamental importance for elucidating the biological processes within a cell. A large proportion of gene regulations happen at the transcriptional level through the binding of transcription factors (TFs) to short regulatory DNA sequences (motifs) in the upstream regions (promoters) of TF target genes. Despite the effort that has been devoted to it, the problem of computationally identifying TF binding sites and motifs (TFBMs) remains a challenge.

Three general families of motif finding approaches have been pursued. Statistical model based approaches are able to derive over-represented degenerate motifs. They usually apply local search strategies to find local multiple alignments that fit some statistical models in order to characterize unknown motifs. Examples of these approaches include Gibbs Sampler (1), MEME (2), Consensus (3) and AlignACE (4). However, these methods typically produce a small number of motifs, can easily get trapped in local minima and are in general too slow to handle large sequences. Word counting based approaches (5,6) identify statistically over-represented motifs through word enumeration. These methods are typically fast and can identify a large number of putative motifs. However, they suffer from the problem of producing too many spurious motifs. A dictionary based approach, recently pioneered by Bussemaker *et al*. in their MobyDick algorithm (7), uses the idea of a dictionary and word usage frequencies to construct sequences. The method progressively considers over-represented words, from short to long. The 'over-representativeness' of a long word is computed as the weighted average of the short words in the current dictionary which can form partitions of the long word. Although this method is, in essence, word counting based, it can filter out many spurious motifs which are over-represented owing to their overlapping with some real motifs. Therefore, it has a higher accuracy than a pure word counting method. However, computing the over-representativeness of a longer word by concatenating shorter ones is problematic, and may miss substantial over-represented motifs.

In our recent research, we approached the problem of motif finding from the perspective of steganography (8). We viewed intergenic sequences as though they formed a stegoscript in which functional TFBMs were secret messages embedded in a covertext of background sequences. Based on this novel viewpoint, we developed an innovative dictionary based motif finding algorithm, which we called WordSpy. We integrated a word counting method and a statistical model into WordSpy. We used the word counting method to examine every possible word and the statistical model to capture over-represented motifs and background words in the given sequences.

*To whom correspondence should be addressed. Tel: +1 314 935 8788; Fax: +1 314 935 7302; Email: zhang@cse.wustl.edu

WordSpy has several favorable features. First, it does not require a background sequence, because it is capable of modeling background words based on the steganographic approach to the problem. This is an important feature for applications where a true background sequence model is hard to determine. Second, WordSpy measures the over-representativeness of a word relative to that of all the other words modeled by the statistical model, resulting in an accurate measure of the over-representativeness of a word. This feature helps to identify motifs of exact length. Third, the algorithm can incorporate gene expression profiling information to separate biologically significant motifs from spurious ones. Fourth, WordSpy is a discriminative motif finding algorithm. It can directly take as input two sets of sequences and find motifs that are over-represented in one set of sequences but not in the other set. This feature makes WordSpy a good algorithm for finding, for instance, tissue-specific TFBMs. Finally, the algorithm can conduct a whole genome analysis on the motifs that it discovers to delineate the fidelity of the motifs to the given sequences. These features can be selected as options, in different combinations, to meet the needs of various motif finding applications such as genome-scale motif finding and discriminative motif finding.

With all these favorable features, WordSpy exhibits superior performance. We validated its performance on ~800 cell-cycle related genes of *Saccharomyces cerevisiae* (9) and sets of genes responsive to 113 TFs which were identified in ChIP-chip experiments (10). We compared WordSpy with the MobyDick algorithm (7) and two popular motif finding algorithms, MEME (2) and AlignACE (4). As shown by our experimental results, available at http://cic.cs.wustl.edu/wordspy/paper, WordSpy significantly outperforms these existing methods.

To make WordSpy freely available to the research community, we developed a web server for the algorithm. In the rest of this paper, we summarize the WordSpy algorithm and discuss the web server interface.

OVERVIEW OF THE ALGORITHM AND ITS FEATURES

We have described the WordSpy algorithm in detail in (11). Here we highlight the main idea and its major steps to help the user understand the capability of the WordSpy web service. To highlight the server parameters discussed here, we put the parameters in italic in the following discussion.

In essence, WordSpy is a deciphering algorithm that treats intergenic sequences as a stegoscript. It learns a dictionary and statistical model to model the stegoscript. The model, shown in Figure 1, can be used to generate a stegoscript as well as to decipher it. In each step of generating a script, a motif word M is chosen with probability P_M, or a background word is selected with probability P_B. When a motif is chosen, a (degenerate) motif W_i is drawn from the motif subdictionary, with probability P_{W_i}, and an exact word w is generated with probability $P(w|W_i)$. Similar is the process for the background word B. This step repeats until the whole script is created.

It is a complex task to learn a model for a stegoscript. We adopt a word counting method to capture motifs of different lengths and progressively build such a model, and apply an

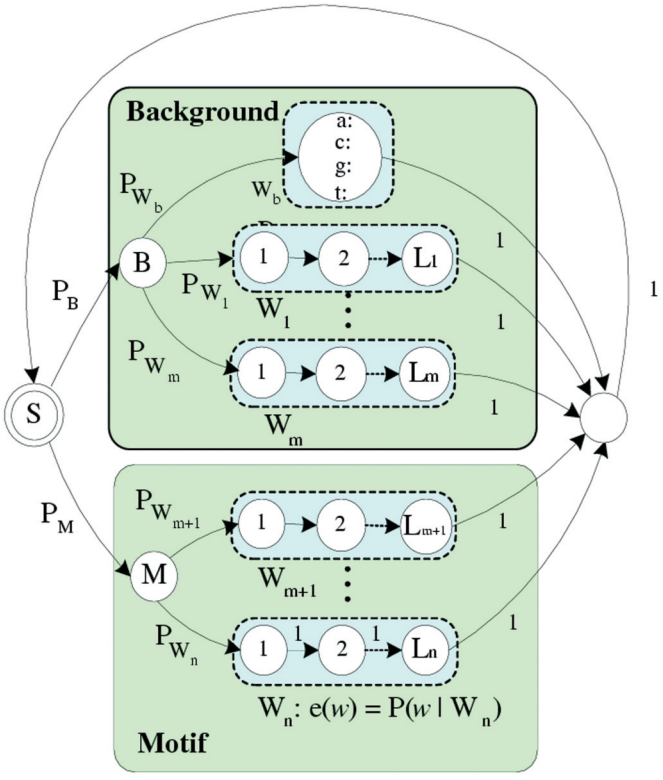

Figure 1. A hidden Markov model for deciphering stegoscripts. It consists of two submodels, the *motif model* and the *background model*. A dashed box represents a word node, which is a combination of several position nodes. Node W_b is a single-base node and always belongs to the background model. States S, B and M do not output any letter.

Expectation Maximization (EM) algorithm (12) to determine the parameters of motif usage. The WordSpy algorithm iterates through three main components. Given a model H_{k-1}, WordSpy first identifies over-represented words of length k. In order to compute the over-representativeness of a word, the given script S is assumed to be generated by H_{k-1}. A word that occurs more often in S than expected is considered over-represented. The user may select an over-representativeness criterion by changing the *word selection ratio*, which is the ratio of word occurrences in S and the expected occurrences generated by H_{k-1}. Some of the newly discovered words are merged by the word clustering component of the algorithm to form degenerate motifs if *degeneracy is allowed*. In this process, motif words will be discriminated from background words using some user specified *motif selection criteria* and will be added to the motif subdictionary. Then, the new model H_k with the new words included will be optimized by the model optimization component of the algorithm. The overall process repeats until the model covers motifs up to the *maximum word length*.

The Z-score is used as the main criterion for selecting high-quality motifs. The Z-score is a standard quantitative measure of over-representativeness. Let N_w be the number of occurrences of a word w in S and let a random variable $\hat{N}_w$ be the number of occurrences of w in the sequences generated by the model H_{k-1}. Denote $E(\hat{N}_w)$ and $\sigma(\hat{N}_w)$ as the mean and standard deviation of $\hat{N}_w$. The Z-score of w is defined as $Z_w = (N_w - E(\hat{N}_w))/\sigma(\hat{N}_w)$. A higher Z-score means that

w is more statistically over-represented. The default minimum *Z*-score threshold for a word is set to 3 in the server.

The user may provide *gene expression data* to improve the quality of predictions. Since co-regulated genes often tend to have similar expression patterns, a motif can be considered biologically meaningful if the genes whose promoters contain the motif have coherent expression patterns. Therefore, WordSpy uses the average pairwise coherence of the expression profiles of the genes associated with a motif to measure its biological significance. This average pairwise coherence is called the *G*-score. When a sufficient amount of gene expression data is supplied, the *G*-score is usually more biologically meaningful than the *Z*-score. However, the *G*-score is also sensitive to the noise in the expression data. The user may consider both measures when selecting motifs. Note that this WordSpy service provides an alternative motif finding strategy which is orthogonal to the conventional method. In the conventional method, co-expressed genes are first selected. This is usually carried out using a gene clustering method, which is often problematic. Motifs are then extracted from the selected genes. In our method, instead of selecting a restricted gene set first, we find putative motifs from a relatively large set of genes. We then choose putative, biologically meaningful motifs based on the expression patterns of the genes containing the motifs. This method avoids the problems associated with gene clustering.

WordSpy is an algorithm for discriminative motifs because of its intrinsic feature of modeling motifs and background words in an integrated model. WordSpy can directly take as input positive and *negative sequence data* and find discriminative motifs that are over-represented in the positive data but not in the negative data. To capture the over-representativeness of a word in the negative sequences, its *Z*-score with respect to the negative sequences, called its *NZ*-score, is also calculated. A discriminative motif should have a high *Z*-score (with respective to the positive sequences) but a low *NZ*-score. This feature makes WordSpy practically useful for many motif finding applications, such as identifying tissue-specific or condition-specific TFBMs and elucidating differential transcriptional regulations. Furthermore, the negative sequences can be treated as background sequences to discriminate biologically meaningful motifs in the positive sequences from background words extracted from the negative sequences. The negative sequences can be the promoters not bound to a certain TF. For instance, we can construct positive and negative data according to ChIP-chip experiment results, e.g. those given in Ref. (10).

A discovered motif should not be considered unique if it also appears very often in promoters arbitrarily chosen from the genome. To prevent such a problem, WordSpy can carry out a whole genome analysis of the specificity of a motif to the promoters among which it was discovered with respect to a set of randomly chosen promoters from the genome. This is done using multiple runs of a Monte Carlo simulation. In each run, a set of promoters are randomly selected from the genome. The number of occurrences of a motif in this set of promoters is determined. A *Z*-score is then computed for the motif to specify how likely the same number of this motif would occur in random promoters. A high positive *Z*-score is desired as it means the motif is likely to be specific to the original input sequences among which it was discovered.

In short, WordSpy is a very flexible, efficient and powerful tool for discovering TFBMs. We have applied it to the promoters of ~800 cell-cycle related genes of *S.cerevisiae* (9). WordSpy is able to identify all known cell-cycle related TFBMs. When the ChIP-chip data of 113 TFs (10) is incorporated, WordSpy can also accurately discover discriminative motifs for each TF. These test results are available at http://cic.cs.wustl.edu/wordspy/paper.

INPUT

The software can be accessed through its web server at http://cic.cs.wustl.edu/wordspy/. WordSpy provides four major services; the first is basic and the remainder are optional. First, it allows the user to discover all over-represented (degenerate) words in a set of DNA, RNA, protein, or English-language sequences. Second, when negative sequences are provided, WordSpy can identify discriminative words that are over-represented in the positive data but not in the negative. Third, when gene expression data are supplied, WordSpy can help select biologically meaningful DNA motifs by calculating a gene expression coherence score, i.e. the *G*-score. The last service is the genome-wide analysis of the quality of putative motifs. The option to use this service is supplied within the result page when the motif result is ready.

The program takes a single sequence or a set of sequences in FASTA format as input. Users can directly input or upload their sequences, or select one from the databases linked to the server. The maximum size of input sequences is currently limited to 500 kilobytes. For a larger input, users can request a local version of the program from the authors.

A user must provide a *maximum word length* and an email address. When the result is ready on the server, the user will be informed where to retrieve the result through an email message. All the other parameters are optional. By default, WordSpy will search for exact words. To find degenerate motifs, the checkbox *allow degeneracy* must be selected. In generating degenerate motifs, WordSpy usually merges the words that have one or two similar over-represented core parts. By checking the box *subtle motifs*, all the words that match at lease *m* bases will be merged, where *m* is determined so that the chance of two random words having *m* base matches is <0.001. Another checkbox is used to specify whether the complementary strands of the input sequences should also be examined.

WordSpy provides several motif selection criteria. *Word selection ratio* gives a criterion for a word to be considered as over-represented. Among the over-represented words, those satisfying the following three criteria will be treated as motif words, whereas the remainder will be treated as background words. These criteria are *minimum Z-score*, *minimum occurrences* and *maximum number of motifs*. A user can also input *gene expression data* and *negative sequence data* to improve the quality of motif predictions.

OUTPUT

The user will be informed via email where to retrieve the motif results when they are ready. Figure 2 shows a screenshot of the result page. The complete result can be downloaded from the

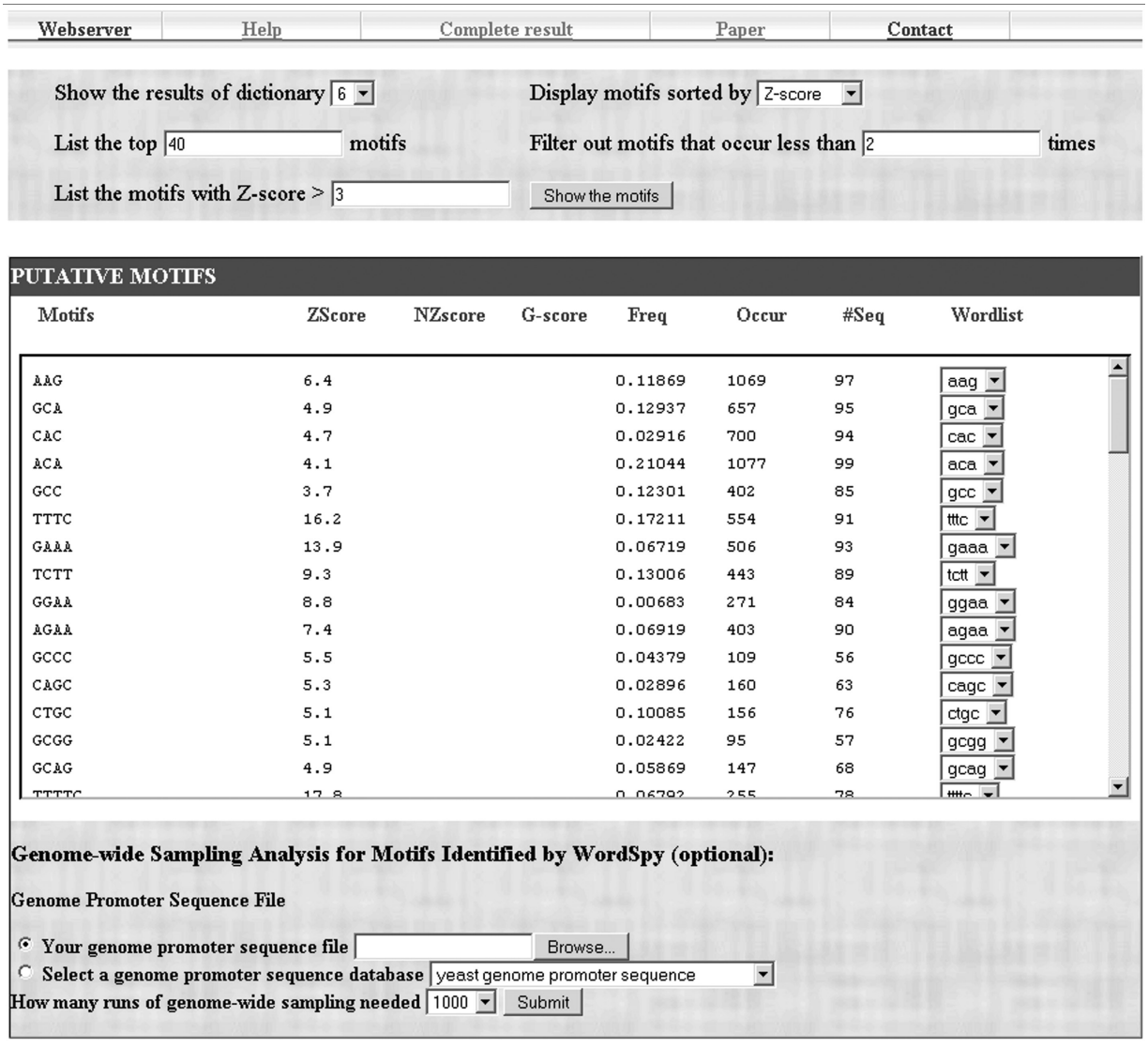

Figure 2. Screenshot of the result page.

Complete result menu at the top. The technical detail of the algorithm and program is accessible through the *Paper* menu. The result page provides a control panel for the user to display the result, with the following.

- Show the results of different dictionaries. A dictionary *k* is an optimized dictionary containing motifs with maximum word length up to *k*.
- Reorder the motifs. The motifs can be reordered by *Z*-score, *NZ*-score, or *G*-score.
- List the top *m* motifs, where *m* is a user input value that specifies the maximum number of motifs to display for each word length.
- Filter out motifs that occur less than *l* times, where *l* is a user input value.
- Filter out motifs that have lower *Z*-score than a user specified value.

The result generated by WordSpy contains a set of putative motifs and a set of deciphered sequences. All the putative motifs in the dictionary up to the maximum word length are listed in the *Putative Motifs* panel. *NZ*-scores specify their over-representativeness in the negative sequences if available. Usually, a higher *Z*-score with a lower *NZ*-score means a better motif. *G*-scores will also be displayed if gene expression data were provided. The user can reorder the list according to the *G*-score through the control panel. *Freq* is the word usage frequency in the statistical model. *Occur* is the number of occurrences of a word in all sequences. *Seq* is the number of sequences that have the motif over-represented. *Wordlist* gives the consensus of the motif. The deciphered sequences are shown in the *Motif Sites* panel, with the motif binding sites highlighted. These deciphered sequences are based on the final optimized model.

The result page also has a control panel for the user to apply a genome-wide analysis to the motifs shown in the *Putative Motifs* panel. The user must provide a *genome promoter sequence file* or select one from the existing promoter database and specify a value for *how many runs of genome-wide sampling needed*. The result is sorted by the *Z*-score in descending order.

IMPLEMENTATION

The WordSpy algorithm was written in ANSI C. The web server was implemented using CGI scripts in PERL. Currently, the web server is hosted on a machine with dual AMD Athlon 1.6 GHz CPUs and 2 GB RAM. Owing to the limited computational power and storage of the current server, it allows only a maximum sequence length of 500 kilobytes. A typical computation on the WordSpy server, searching 500 kilobytes of sequences with default parameter settings, takes ~15 min. The link to the output page will be sent to the user through email when available. A local copy of the software can analyze sequences up to 10 megabytes with a reasonable amount of time.

ACKNOWLEDGEMENTS

This research was supported in part by NSF grants IIS-0196057 and EIA-0113618 to W.Z. Funding to pay the Open Access Publication charges for this article was provided by U.S. National Science Foundation.

Conflict of interest statement. None declared.

REFERENCES

1. Lawrence,C.E., Altschul,S.F., Bogouski,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.
2. Bailey,T.L. and Elkan,C. (1995) Unsupervised learning of multiple motifs in biopolymers using EM. *Machine Learning*, **21**, 51–80.
3. Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.
4. Hughes,J.D., Estep,P.W., Tavazoie,S. and Church,G.M. (2000) Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.*, **296**, 1205–1214.
5. van Helden,J., Andre,B. and Collado-Vides,J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, **281**, 827–842.
6. Sinha,S. and Tompa,M. (2000) A statistical method for finding transcription factor binding sites. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, 19–23 August, Price Center, University of California San Diego, La Jolla, CA, pp. 344–354.
7. Bussemaker,H.J., Li,H. and Siggia,E.D. (2000) Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. *Proc. Natl Acad. Sci. USA*, **97**, 10096–10100.
8. Wayner,P. (2002) *Disappearing Cryptography*, 2nd edn. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
9. Spellman,P.T., Sherlock,G., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.
10. Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.
11. Wang,G. and Zhang,W. (2005) An iterative learning algorithm for deciphering stegoscripts: a grammatical approach for motif discovery. Washington University, Department of Computer Science and Engineering, Technical Report No. 12, 2005. St. Louis, MO, USA.
12. Dempster,A.P., Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R. Stat. Soc.*, **39**, 1–38.