cWINNOWER Algorithm for Finding Fuzzy DNA Motifs

Shoudan Liang
NASA Advanced Supercomputing Division
NASA Ames Research Center
Moffett Field, CA, 94035 USA
email:sliang@nas.nasa.gov
telephone: (650) 604 6631
fax: (650) 604-0987

(Dated: January 29, 2003)

Abstract

The cWINNOWER algorithm detects fuzzy motifs in DNA sequences rich in protein-binding signals. A signal is defined as any short nucleotide pattern having up to d mutations differing from a motif of length l. The algorithm finds such motifs if multiple mutated copies of the motif (i.e., the signals) are present in the DNA sequence in sufficient abundance. The cWINNOWER algorithm substantially improves the sensitivity of the winnower method of Pevzner and Sze by imposing a consensus constraint, enabling it to detect much weaker signals. We studied the minimum number of detectable motifs q_c as a function of sequence length N for random sequences. We found that q_c increases linearly with N for a fast version of the algorithm based on counting three-member sub-cliques. Imposing consensus constraints reduces q_c by a factor of three in this case, which makes the algorithm dramatically more sensitive. Our most sensitive algorithm, which counts four-member sub-cliques, needs a minimum of only 13 signals to detect motifs in a sequence of length $N = 12000$ for $(l, d) = (15, 4)$.

KEY WORDS: DNA motif, winnower, transcription factor binding signals.

I. INTRODUCTION

Transcription factors binding to fuzzy motifs in DNA is a process that underlies one of the most important modes of gene regulation in a cell. Algorithms that identify such protein-binding signals in DNA are becoming especially important with the recently developed high-throughput techniques that show promise to uncover these interactions at a genome-wide scale. One such technique is genome-wide location analysis[1, 2], in which DNA microarrays offer a means to identify the approximate locations of the binding sites of a transcription factor anywhere in the genome. Motif identification also proves useful for analyzing microarray data when the measured mRNA expression levels of all the genes on the microarray are clustered, and genes in the same cluster are assumed to contain common regulatory elements in the DNA upstream of the transcription initiation points. Then the bioinformatics problem is to find common motifs in the upstream of the clustered—and presumably co-regulated—genes that may play similar regulatory roles for each[3, 4]. Motif identification should also aid the search for morphogen binding sites from enhancers controlling early embryo development. Large-scale mapping of such *cis*-regulatory signals has been proposed[5] in which random genomic DNA sequences are assessed for their potential regulatory role in early embryo development. Detailed studies on sea urchin embryos have demonstrated that such enhancers often contain multiple transcription factor binding sites (34 have been identified in the case of *endo16*)[6]. Fuzzy motif-finding algorithms may prove to be a general method for identifying transcription factor binding sites from the large-scale mapping of *cis*-regulatory elements.

In a typical situation, a weak DNA signal is embedded in a set of experimentally identified DNA sequences that are enriched with the binding site. Not every sequence in the experimental set is guaranteed to contain the binding site. In addition, protein-binding DNA signals often contain ambiguous positions which can have more than one equivalent nucleotide. The bioinformatics problem is to determine the hidden signal, if any. Identifying such a weak signal is a non-trivial problem[7]. Pevzner and Sze[8] have formulated a 'grand challenge' problem of finding a hidden motif of length l in which each binding site can differ from the hidden motif in at most d places. Two length l motifs can share a common hidden pattern even when they differ in $2d$ places. If a graph is constructed where the nodes are consecutive length l patterns and two nodes are linked if two corresponding patterns differ in no more than $2d$ places, then in the set of patterns that are within Hamming distance d to the common motif, every node is connected to every other node. In other words, the nodes form a q-member clique (q-clique) where q is the size of the set. However, the graph contains vastly more spurious connections. Pevzner and Sze proposed a winnower method that systematically deletes spurious links that cannot be a part of the q-clique.

The chief advantage of the winnower algorithm is that it is guaranteed to find *all* hidden patterns that are present at least q times in the sample DNA sequences within Hamming distance d from the hidden pattern. The hidden pattern itself does not have to even be present in the dataset. Most popular motif finding methods [10–15] rely on optimization of non-linear objective functions and therefore cannot guarantee that the pattern found attains the global optimum. The winnower method is unique in being able to claim the definitive absence of signal (l, d) for copy number q.

In this paper, we introduce a consensus bound on patterns belonging to the same clique that enables the algorithm to remove more spurious links. As a result, the algorithm becomes substantially more sensitive. We discovered parameter regions where the winnower method is

TABLE I: *The table compares the smallest copy number q necessary to prune all spurious links for one random sequence of length N (roughly equivalent to q sequences of length $\frac{N}{q}$ each considered by Pevzner and Sze[8]). WINNOWER and cWINNOWER denotes winnower algorithms without and with consensus bounds. Both cases are listed for the pruning criteria base on eliminating links by counting the number of 3-cliques ($k = 2$) and 4-cliques ($k = 3$). As q increases, the percentage of links pruned changes from essentially 0% to 100% in a very narrow range. The q_c listed in this table is the smallest q when the number of spurious links left unpruned is less than 1000 (the total number of spurious links is more than a million). For reasons explained in the text, the algorithms run slowly in the vicinity of q_c. For $N = 12000$ the running time for the WINNOWER algorithm for q in the range indicated exceeds 48 hours. Only a range of values is given in this case.*

Sequence Length N	3000	6000	12000
$k = 2$ WINNOWER	18	35	71-76
$k = 2$ cWINNOWER	10	13	23
$k = 3$ WINNOWER	11	15	23-32
$k = 3$ cWINNOWER	8	10	13

effective in detecting signals. We computed minimum clique sizes, q_c, required for removing all spurious links generated from a random sequence. Our main results are summarized in Table I. We find that q_c increases approximately linearly with the length of the random sequence for the case of $k = 2$ (that eliminates links by counting the member of 3-cliques). The consensus constraint reduces q_c by a factor of three for $k = 2$. For our most sensitive case of $k = 3$(that counts 4-cliques), which is slower to run, $q_c = 13$ for the longest sequence length we tried ($N = 12000$). This is about a factor of two better than without consensus constraints. We formulated the algorithm in terms of set operations resulting in a much simpler implementation. For the most sensitive case of $k = 3$, which usually runs too slow to be useful, we speeded up the calculation by saving certain intermediate results. For the longest sequences we considered ($N = 12000$ and $(l, d) = (15, 4)$), the $k = 3$ algorithm is only a factor of three slower than $k = 2$.

We first review the winnower algorithm before proving a consensus constraint. We then present the cWINNOWER algorithm in terms of set operations and discuss tests on random sequences.

II. WINNOWER METHOD

Imagine the hyperspace of all possible length l patterns populated by words cut consecutively from the input sequences. Regions of the hyperspace that have an unusually high density of l-words indicates statistical significance and presumably biological meanings. Theoretically, we can enumerate each sequence and count the number of sequences that are within d mutations from it. However, the computation required goes up exponentially with the length of the pattern and becomes impossible for moderately long patterns. The winnower method[8] on the other hand changes the finding of hidden motifs to a graph-theoretical problem.

Define a graph of n nodes each of which represents a consecutive sub-string of input sequences. Two nodes are connected if they differ in less than $2d$ positions. The crucial observation is that all the sub-strings sharing a common motif form a clique graph. (A q-clique graph has q nodes and every pair of nodes is connected.)

Most of the connections made by the $2d$ mismatch criterion are spurious, in that they are not part of a graph that makes a q-clique, and so must be eliminated. For example, any node that has less than $(q-1)$ connections to the rest of the graph cannot be a part of the q-clique, and these connections are all spurious. Furthermore, in order for a link to be a part of the q-clique, it must be a part of at least $(q-2)$ triangles. More specifically, suppose the link is between node a and node b. Define a triangle neighbor set C consisting of nodes c that are connected to both a and b. The number of nodes in C has to be larger than $(q-2)$. This triangle criteria is what Pevzner and Sze called the $k=2$ case[8]. A more stringent test is the $k=3$ case, in which each member of set C is required to be in four-member sub-cliques. In order to belong to set C, there must be at least $(q-3)$ 4-cliques containing nodes a, b and c. Finally, the size of the set C must be larger than $(q-2)$. This easily generalizes to higher clique graphs.

III. CONSENSUS CONSTRAINT

By counting the number of 3-clique and 4-clique sub-graphs, the winnower method systematically eliminates links that cannot be a part of the q-clique. The main difficulty is that the $2d$-mismatch criterion that defines a connection for the link is too lenient, resulting in an explosion of spurious links. Here we develop a consensus constraint for deciding if a link between a and b belongs to a q-clique. Let P be the set of nodes including a and b as well as the nodes connected with both a and b. Let $n=|P|$ be the number of nodes in set P. In order for the link (a,b) to be a part of the q-clique, we must have $n \geq q$. In order for the link to belong to the q-clique, the following consensus constraint must be satisfied:

$$\sum_{i=1}^{l} \min(q, \sum_{j=1}^{n} \delta_{S_i^j, C_i}) \geq q(l-d) \tag{1}$$

where S_i^j is the i-th element of the j-th sequence in P. C_i is the i-th column consensus of the sequences S^j, *i.e.*

$$\sum_{j=1}^{n} \delta_{S_i^j, C_i} = \max(\sum_{j=1}^{n} \delta_{S_i^j, A}, \sum_{j=1}^{n} \delta_{S_i^j, C}, \sum_{j=1}^{n} \delta_{S_i^j, G}, \sum_{j=1}^{n} \delta_{S_i^j, T})$$

We now prove the consensus constraint. In order for the set P of n sequences filtered by either $k=2$ or $k=3$ criteria to be a part of the q-clique, we must have

$$P_{>} = \left\{ S \mid S \in P, \sum_{i=1}^{l} \delta_{S_i, h_i} \geq l-d \right\} \tag{2}$$

$$|P_{>}| \geq q \tag{3}$$

for some hidden motif h. Here $\{.\}$ denotes a set and $|.|$ is the number of elements in it. l is the length of the string, and d is the maximum number of allowed mismatches.

This equation implies

$$\sum_{j=1}^{q} \sum_{i=1}^{l} \delta_{S_i^j, h_i} \geq q(l-d) \tag{4}$$

for at least one subset of q sequences in $P_>$. If we can show no q sequences satisfy Eq.(4), then Eq.(3) fails.

Consider the consensus sequence of all sequences in $P_>$. By definition, the consensus sequence C maximizes the sum $\sum_{j=1}^{n} \delta_{S_i^j,C_i}$ for each position i, where $n = |P_>|$.

Obviously,

$$\begin{aligned}
\sum_{j=1}^{q}\sum_{i=1}^{l} \delta_{S_i^j,h_i} &= \sum_{i=1}^{l}\sum_{j=1}^{q} \delta_{S_i^j,h_i} \\
&\leq \sum_{i=1}^{l}\sum_{j=1}^{q} \delta_{S_i^j,C'_i} \\
&\leq \sum_{i=1}^{l} \min(q, \sum_{j=1}^{n} \delta_{S_i^j,C_i})
\end{aligned}$$

where C' is the consensus sequence of q sequences and C is the consensus of the whole group of n sequences that include q sequences as a subset. The last inequality follows because

$$\begin{aligned}
\sum_{j=1}^{q} \delta_{S_i^j,C'_i} &= \max(\sum_{j=1}^{q} \delta_{S_i^j,A}, \sum_{j=1}^{q} \delta_{S_i^j,C}, \sum_{j=1}^{q} \delta_{S_i^j,G}, \sum_{j=1}^{q} \delta_{S_i^j,T}) \\
&\leq \min(q, \max(\sum_{j=1}^{n} \delta_{S_i^j,A}, \sum_{j=1}^{n} \delta_{S_i^j,C}, \sum_{j=1}^{n} \delta_{S_i^j,G}, \sum_{j=1}^{n} \delta_{S_i^j,T})) \\
&= \min(q, \sum_{j=1}^{n} \delta_{S_i^j,C_i})
\end{aligned}$$

A useful special case is when pruning a link that has the maximum allowed mismatch $2d$. The majority of links will be of this type. For such a link, the positions of matched nucleotides must agree with the consensus sequence. In such a case, using the nucleotides from the matching part of the two nodes instead of the n sequence consensus will improve the bound.

We have also derived other constraints for three nodes. However, none proved as useful in practice as the consensus bound.

IV. CWINNOWER ALGORITHM

Patterns l nucleotides long cut consecutively from each of the input sequences form a set of nodes. Any two nodes are linked if their mismatches are less than $2d$, because they can be within d-mutations from a common hidden pattern.

The winnower method systematically removes spurious edges that cannot be a link in a q-clique because they lack a sufficient number of sub-cliques. Here we give some details of the procedures used: $k = 1$ counts the number of links to each node; $k = 2$ counts the number of triangles (3-cliques) for a given link; and $k = 3$ counts 4-cliques.

A. k=1 pruning criterion

If the number of links, n, to any node is smaller than $q - 1$, then the node cannot be a part of any q-clique. Prune all links connected to it. If n is smaller than $4(q-d)$, then apply the consensus criteria. If it fails to satisfy, prune all links. The algorithm has a complexity of $O(N^2)$, where N is the total length of the sequence.

B. k=2 pruning criterion

k=2 pruning decides if the link between a and b should be pruned. First, define a set P of all nodes c that are connected to both a and b. Next apply the consensus criterion with $n = 3$ and $q = 3$ on a, b and c to remove those that fail the criterion from P. Finally, if the size of the neighboring set $|P|$ is smaller than $q - 2$, then the node cannot be in any q-clique. Prune the link between a and b. If $|P|$ is smaller than $4(q - d)$ (beyond this value the consensus bound becomes useless), then apply the consensus criteria. If it fails to satisfy, prune the link (a, b). The algorithm has a complexity of $O(N^3)$.

C. k=3 pruning criterion

The goal is to prune the link between a and b. Here we also have a neighboring set P. The criterion of admission to the set is more stringent. Each member of the set P must have at least $q-3$ other nodes that together with a, b and c form $(q-3)$ 4-cliques. In order for the four nodes to form a 4-clique, they must satisfy the consensus criterion with $n = 4$ and $q = 4$. Finally, the size of the set P must be larger than $q - 2$. The consensus criterion is applied to P.

By rearranging the order of calculation and saving some intermediate results, the code can be made much faster. Let N_t denote the set of neighbors connected to node t. Define $I_b = N_a \bigcap N_b$ for each b in N_a. Notice that I_b are typically much smaller than N_a because it is the intersection of two sets. This set of sets will be used repetitively in pruning links between a and its neighbors b in set N_a. To determine if any $c \in I_b$ belongs to set P, we form $J_c = I_c \bigcap I_b$ and require that the size of J_d be larger than or equal to $q - 3$. The complexity of the algorithm is $O(N^4)$. However, there is a much larger $O(N^3)$ term comparable to the $k = 2$ case. The running time for $q = 32$, $(l, d) = (15, 4)$ and $N = 12000$ for $k = 2$ is 3 hours 45 minutes on a SGI workstation. This is to be compared with a running time of 10 hours and 15 minutes for $k = 3$ at $q = 18$.

In order to speed up computations, we prune the node with the smallest numbers of links first because they are the easiest to be pruned. This will have a domino effect on other nodes resulting in a shorter running time. The extra cost is a simple sorting. By the same token, we prune links with $2d$ mismatches first. The algorithm is implemented in C++ using the standard template library, which has fast set operations.

V. PRUNING RANDOM SEQUENCES

Because the winnower algorithm is guaranteed not to prune away true q-cliques, its ability to remove spurious links determines its performance. In a typical situation, almost all of the links are spurious. For example, for $(l, d) = (15, 4)$, and total sequence length $N = 6000$ the total number of links is about one million with the $2d$ mismatch criterion, whereas embedding twenty signals only contributes 190 links.

In this paper we perform tests on random sequences. This is preferable to testing on random sequences embedded with signals[8] because, in our experience, on random sequences

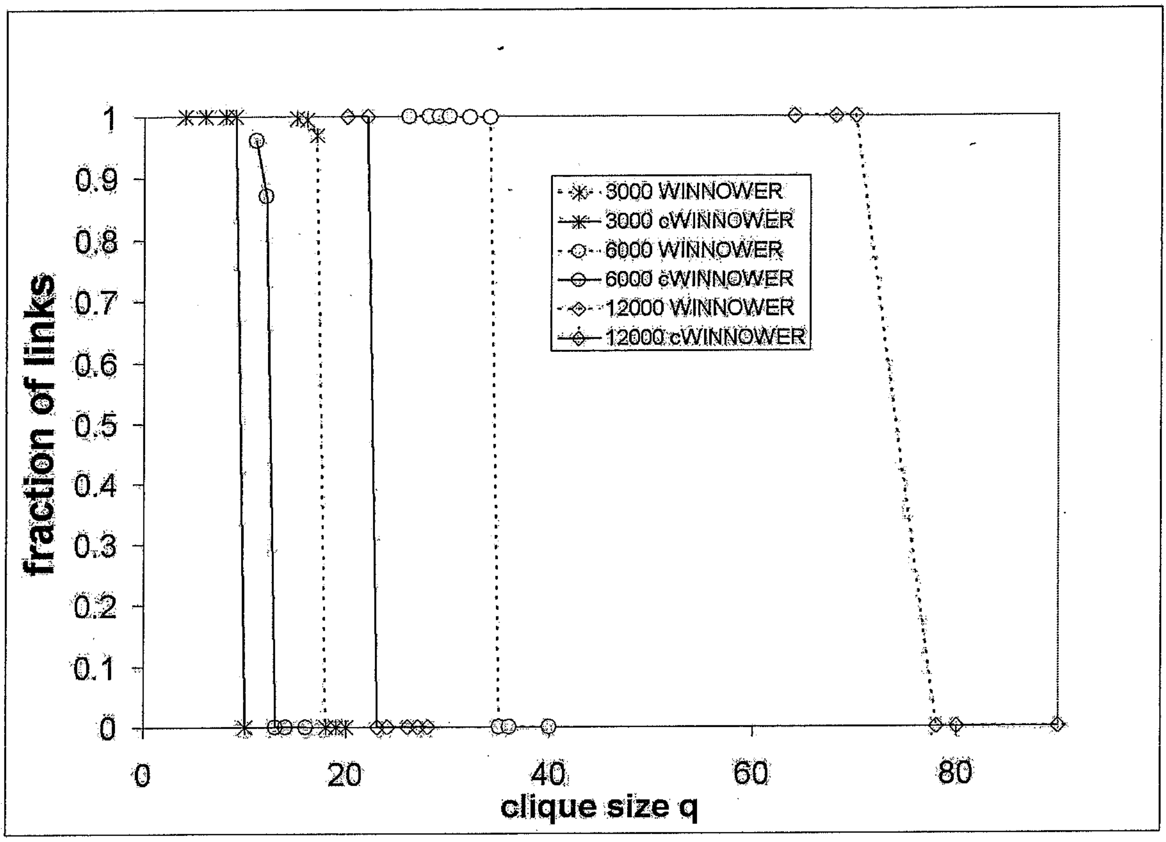

FIG. 1: Fraction of spurious links left unpurged as a function of clique size q for the case of $k = 2$ after WINNOWER (dashed lines) and cWINNOWER (solid lines) algorithms are applied to random sequences of three sizes: N=12000 (diamonds); N=6000 (circles); and N=3000 (stars). cWINNOWER uses consensus constraints whereas WINNOWER does not. In each case, there is an abrupt transition from one (no link is purged) to zero (all links are purged). For WINNOWER algorithm $N = 12000$ (the rightmost curve), there are no data for q between 71 and 75 because it takes too long to complete the calculations. The hidden pattern sought is $(l, d) = (15, 4)$, *i.e.*, a pattern of length 15 with at most 4 mutations allowed. Each point is the result of averaging between one to ten random sequences.

embedded with a pattern the winnower algorithm often leaves more links unpruned (by a factor of two to three) even when the algorithm can remove all spurious links for the random sequences of the same length without embedded signals. The last few hundred spurious links mixed with links from true signals can be dealt with easily by other methods of finding clique graphs.

The number of spurious links per node increases linearly with the total sequence length. Longer sequences require a larger q. We determine the minimum $q = q_c$ as a function of total sequence length for pattern $(l, d) = (15, 4)$. Algorithms that can detect signals with small copy number q are more sensitive. FIG.1 shows the percentage of spurious links left unpruned as a function of q after the application of winnower algorithms with and without consensus bounds for the case of $k = 2$ (which counts number of 3-cliques to eliminate spurious links).

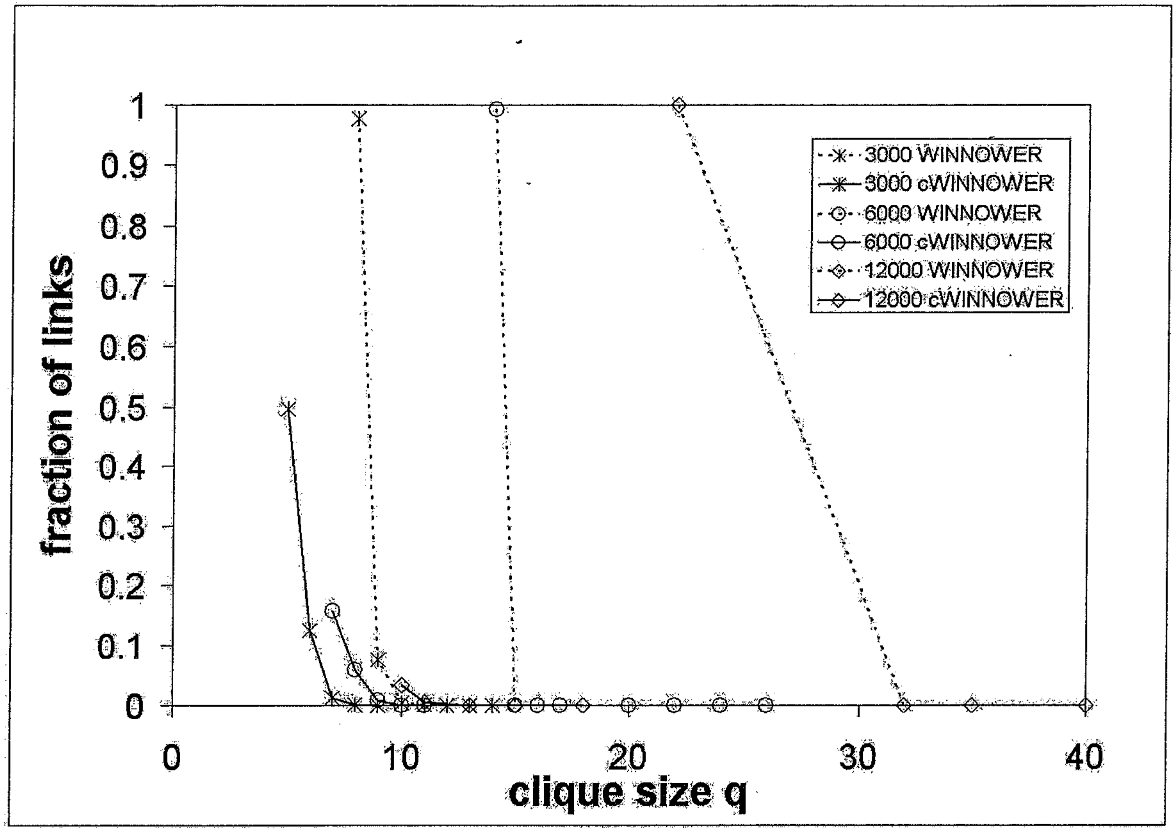

FIG. 2: Same as Fig.1 but for the case of $k = 3$. Transitions are in general not as sharp as in the case of $k = 2$. For the rightmost curve, for q between 23 and 31, the calculation takes too long to complete.

The calculations were performed for random sequences of lengths $N = 3000, 6000$ and 12000. For each length, as the copy number q increases, a sharp transition is seen which defines the minimum copy number q_c. For $q < q_c$, the algorithm is unable to prune all the spurious links and therefore unable to find the q-clique containing the signals. We have also performed similar calculations for the more sensitive case of $k = 3$(FIG.2), which eliminates links by counting the number of 4-cliques. Table I lists q_c for three random sequence lengths for the winnower algorithm with and without consensus bounds. The minimum detectable copy number q_c is much smaller with the consensus bounds. For the case of $k = 2$, q_c increases linearly with the random sequence length. (This is clearly true for WINNOWER and is most likely also true for cWINNOWER.) The minimum detectable copy number q_c is three times smaller with the consensus constraint than without for $k = 2$. Therefore the consensus constraint greatly improved the sensitivity of the algorithm. For the more sensitive $k = 3$ algorithm, q_c also become smaller when the consensus bound is imposed. The ratio of q_c with consensus constraints to q_c without consensus constraints increases with the sequence length (see Table I).

For $N = 12000$, there is a range of q near q_c where the calculation took too long to run. The reason is that the program stops when it goes through all links remaining and there are no more links to delete. However, near q_c, each time it goes through the list, it is able to delete a few links but the bulk of computing time is spent on deciding whether to delete links that end up not being deleted.

Calculations reported in FIG.1 and FIG.2 were averaged between one and ten random sequences depending on the sequence length. Where the calculations were repeated on different random sequences of the same length, the results are always similar, which is expected because the number of spurious links is very large: 0.25×10^6, 10^6, and 4×10^6 for $N = 3000, 6000$ and 12000, respectively.

Winnower assumes mutations can occur anywhere in the sequence. However in most of the regulatory sequences, some sites are more conserved than others. A partial remedy for this is to divide the pattern into two parts. For one part—for example, in the middle of the pattern—the allowed number of mutations is less than that for the rest of the pattern.

In summary, the winnower method affords several advantages. In addition to being able to allow mutations in the hidden pattern, it has a clear-cut criterion for signal selection and is also unique in showing the absence of signal, *i.e.* it can prove that a certain (l, d) motif occurs less than q times in the input sequence.

VI. ACKNOWLEDGEMENTS

I thank Profs. Bob Laughlin (Stanford University), Hao Li (University of California San Francisco), and Zhifeng Shao (University of Virginia) for numerous stimulating and informative discussions. I also thank Kevin Harrington for a critical reading of the manuscript.

[1] Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings EG, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, Volkert TL, Wilson CJ, Bell SP, Young RA. (2000) Genome-wide location and function of DNA binding proteins. *Science* **290** 2306-9.

[2] Iyer VR, Horak CE, Scafe CS, Botstein D, Snyder M, Brown PO. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* **409**, 533-8.

[3] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. *Nat. Genet.* (1999) **22**, 281-5.

[4] Zhu Z, Pilpel Y, Church GM. (2002) Computational Identification of Transcription Factor Binding Sites via a Transcription-factor-centric Clustering (TFCC) Algorithm. *J. Mol. Biol.* **318**, 71-81.

[5] Harafuji N, Keys DN, Levine M. (2002) Genome-wide identification of tissue-specific enhancers in the Ciona tadpole. *Proc Natl Acad Sci U S A.* **99**, 6802-5.

[6] Davidson, E. Genomic Regulatory Systems, Development and Evolution (Academic Press 2001)

[7] for a recent review, see for example Li H. (2002) Computational approaches to identifying transcription factor binding sites in yeast genome. *Methods Enzymol.* **350**:484-95.

[8] Pevzner P.A. and Sze S.-H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. Proc. of 8th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'2000), 269-278.

[9] Sze S.-H., Gelfand M.S. and Pevzner P.A. (2002) Finding weak motifs in DNA sequences. Pacific Symposium on Biocomputing (PSB'2002), 235-246.

[10] Lawrence, C.E. Altschul, S. F. Boguski, M. S., Liu, J. S., Neuwald, A. F., Wooton, J. C.,(1993) *Science*, **262**, 208 ; Liu, J. S., Neuwald, A. F., and Lawrence, C.E., (1999) *J. Am. Statist. Assoc.* **90** 1156 .

[11] Bailey, T.L. and Elkan, C., (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers *Proc. of 8th Int. Conf. on Intelligent Systems for Molecular Biology* pp. 28-36, AAAI Press, Menlo Park, California,

[12] Hertz G.Z. and Stormo G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563-77.

[13] Bussemaker, H.J., Li, H. & Siggia, E.D. (2000) Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. *Proc Natl Acad Sci U S A.* **97** 10096-10100 .

[14] Liu, X. Brutlag, D.L. & Liu, J. S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.* 127-138 .

[15] Liu XS, Brutlag DL, Liu JS. (2002) An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nat. Biotechnol.* **20**, 835-9.