

Identifying regulatory signals in DNA-sequences with a non-statistical approximation approach [⌘]

Cun-Quan Zhang ^y

Department of Mathematics

P. O. Box 6310

West Virginia University

Morgantown, WV. 26506

Email: cqzhang@math.wvu.edu

Yunkai Liu and Elaine M. Eschen

Lane Department of Computer Sciences and Electronic Engineering

P. O. Box 6109

West Virginia University

Morgantown, WV. 26506

Email: liuyk_chi@yahoo.com and eeschen@csee.wvu.edu

Keqiang Wu ^z

Department of Biology

P. O. Box 6057

West Virginia University

Morgantown, WV. 26506

Email: kewu@mail.wvu.edu

Abstract

The identification of regulatory signals is one of the most difficult and challenging tasks in bioinformatics. With the development of microarray technology, biologists can now reveal gigantic gene sequences, which contain parts of the genome believed to be responsible for most transcription

[⌘]Keywords: approximation algorithm, string algorithm, graph algorithm, motif searching, clique searching, representative, R-group, similarity, Hamming distance, Menhaden distance, TFBM

^yPartially supported by National Security Agency under Grant MDA904-01-1-0022

^zPartially supported by US Department of Agriculture under Grant 2002-35301-12208

control - the transcription factor DNA-binding motifs (TFBMs). Developing a practical and efficient computational tool to identify TFBMs will enable us to better understand the interplay among thousands of genes in a complex eukaryotic organism.

This optimization problem is mathematically formulated as the motif finding problem in computer science. This problem has been studied extensively in recent years. In this paper, we have developed a new mathematical model and approximation technique for motif searching, in which graph theoretic and geometric properties of this approach are applied. Based on the properties of this model, we propose a non-statistical approximation algorithm to find motifs in a set of genome sequences.

1 Introduction

The identification and interpretation of the regulatory signals within the eukaryotic genomes remain among the greatest goals and most difficult challenges in genome research. With the development of gene-profiling technology, the expression patterns of thousands of genes under a variety of conditions have been revealed. This gives us the opportunity to identify and analyze the parts of a genome believed to be responsible for most transcription control - the transcription factor DNA-binding motifs (TFBMs). Developing bioinformatics tools to identify TFBMs in the eukaryotic genome will be very useful. These tools will lead to a better understanding how the interplay among thousands of genes leads to the existence of a complex eukaryotic organism. Understanding gene regulation is one of the most exciting topics in molecular genetics. The quantity of information gained in the sequencing and gene expression projects both requires and enables us to use computers to solve this problem. We are developing bioinformatics tools to identify the TFBMs in the genomes. The arrival of microarray gene-expression data has generated a large group of genes with a similar expression profile (e.g. those that are activated at the same time in the cell cycle). This gene expression profile is, at least partly, caused by and reflected in a similar structure of the regions involved in transcription regulation. The ultimate goal is the automated construction of specific promoter models containing a combination of several TFBMs. Another approach is to come up with algorithms able to identify the TFBMs in genomes.

There are many papers about identification of motifs in the literature and related software, which are motivated by not only genome research but also data mining and other areas. Smith and Waterman are among the early pioneers who studied the motif finding problem and introduced local dynamic programming [11]. FASTA [10] is another practical approach for alignments. The suffix tree [7] method of Gusfield provides optimal solutions for motifs consisting of identical substrings. For searching motifs that allow a small amount of mutations, a greedy algorithm [8] is efficient. Lawrence, et al. developed an algorithm based

on Gibbs-sampling [9]. Another technique that has been used to solve the motif finding problem is "expectation maximization" [3].

Let S be a set of strings with the same length n . The goal of our project is to find a collection of subsets (motifs) of S in each of which elements are very "similar" to each other. Akutsu [1] [2] proved the motif finding problem (measured by relative entropy) is NP-complete. In this paper, we establish a mathematical model: the set of strings/data is mapped to a graph in which each vertex represents a string and the weight of the edge between two vertices is the similarity between the corresponding strings. Obviously, it is a complete graph $K_{|S|}$ with the weight function $\hat{A} : E(K_{|S|}) \rightarrow [0; 1]$. The motif finding problem is obviously equivalent to the problem of "finding all cliques" in a graph where only edges with large weight are present. It is well-known that the maximum clique problem is NP-complete [5]. (In fact, the number of large cliques in some graphs could be exponential.) This means it is not likely that a polynomial time algorithm exists for an optimal solution of this problem if only graph theoretic methods are applied.

We investigate some properties (such as, the "triangle inequality") about our model in Section 3. Based on these geometric and graph theoretic properties, we are able to introduce a new approximation technique for motif searching (or clique-like subgraph searching in weighted graphs).

2 Definitions and Mathematical Model

In this section, a similarity function between strings corresponding to Hamming distance is defined. All properties in this paper are proved based on Hamming distance, although many of them are presented in terms of the similarity function. A mathematical model is to be established based on those definitions.

Graph theory notation and terminology used in this paper are standard in most discrete mathematics and computer science textbooks [4], [13], etc.

2.1 Definitions

DNA sequences consist of four types of nucleotides, named Adenine, Cytosine, Guanine and Thymine. Hence, DNA sequences can be presented as strings over the alphabet $\mathcal{S} = \{A; C; G; T\}$. For the sake of convenience, we denote $A = z_1; C = z_2; G = z_3; T = z_4$. If there are two strings $S_1 = x_1; \dots; x_n$ and $S_2 = y_1; \dots; y_n$, let $\pm(x_j; y_j) = 1$ if $x_j = y_j$, and $\pm(x_j; y_j) = 0$ otherwise. (If protein sequences are in consideration, we may simply replace the nucleotide alphabets of order 4

with the amino acid alphabets of order 20).

Definition 2.1 (Representative of a set of strings) Let S be a set of strings with the same length n . The representative of S , denoted by $R(S)$, is a $(4 \times n)$ -matrix $[u_{i,j}]$ such that $u_{i,j}$ is the frequency (%) of the symbol 2_i in the j -th position of all strings of S . $\sum_{i=1}^4 [u_{i,j}] = 1$: And we say the number of strings of S as $|S|$.

Definition 2.2 (Hamming Distance between strings) The Hamming Distance of two strings $S_1 = x_1; \dots; x_n$ and $S_2 = y_1; \dots; y_n$ is defined as follows,

$$d_H(S_1; S_2) = \sum_{j=1}^n (1 - \delta(x_j; y_j)):$$

Or, it can be equivalently defined as: Let $D \subseteq \{1; \dots; n\}$ such that $x_i = y_i$ if and only if $i \in D$. Then $d_H(S_1; S_2) = |D^c|$.

Definition 2.3 (Similarity between strings) The Similarity of two strings S_1 and S_2 with the same length n is defined as follows,

$$\hat{A}(S_1; S_2) = 1 - \frac{d_H(S_1; S_2)}{n} = \frac{1}{n} \sum_{j=1}^n \delta(x_j; y_j):$$

Or, it can be equivalently defined as: Let $D \subseteq \{1; \dots; n\}$ such that $x_i = y_i$ if and only if $i \in D$. Then $\hat{A}(S_1; S_2) = \frac{|D|}{n}$.

Definition 2.4 (Hamming Distance and Similarity between string and representative) Let S be a set of strings with the same length n and $R(S) = [r_{i,j}]_{4 \times n}$ be the representative of S . Let $X = x_1 \dots x_n$ be a string of length n (X is not necessarily in S). The Hamming Distance between X and $R(S)$ is defined as follows,

$$d_H(R(S); X) = \sum_{j=1; \dots; n; \text{ and } x_j = ^2_i}^n (1 - r_{i,j}) = \sum_{j=1}^n \sum_{i=1}^4 r_{i,j} \delta(1 - \delta(x_j; ^2_i)):$$

We also define the Similarity between X and $R(S)$ as follows,

$$\begin{aligned} \hat{A}(R(S); X) &= 1 - \frac{d_H(R(S); X)}{n} \\ &= \frac{1}{n} \sum_{j=1; \dots; n; \text{ and } x_j = ^2_i}^n r_{i,j} \\ &= \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^4 r_{i,j} \delta(\delta(x_j; ^2_i)) \end{aligned}$$

Definition 2.5 (Hamming Distance and Similarity between representatives) Let S and S^0 be two sets of strings with the same length n and $R(S) = [u_{ij}]$ and $R(S^0) = [v_{ij}]$. The Hamming Distance between $R(S)$ and $R(S^0)$ is defined as follows,

$$d_H(R(S); R(S^0)) = \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n |u_{ij} - v_{ij}|$$

We also define the Similarity between $R(S)$ and $R(S^0)$ as follows,

$$\hat{A}(R(S); R(S^0)) = 1 - \frac{d_H(R(S); R(S^0))}{n} = \frac{1}{2n} \sum_{j=1}^n \sum_{i=1}^n |u_{ij} - v_{ij}|$$

Definition 2.6 A c -clique H of an edge-weighted graph G is a subgraph of G such that $w(u;v) \geq c$ for every edge $u;v \in E(H)$.

Definition 2.7 Let G be an edge-weighted graph and G^0 be a subgraph of G . The subgraph G^0 is called an R -group with similarity c if $\hat{A}(x; R(G^0)) \geq c$, for every $x \in G^0$. R -group can be defined equivalently as follows. G^0 with k^0 strings is an R -group if,

$$\frac{1}{k^0} \sum_{y \in G^0} \hat{A}(x; y) \geq c;$$

for every $x \in G^0$.

Obviously, R -groups are approximations of c -cliques. It is not hard to see that a c -clique Q is an R -group since the similarity between every pair of vertices (strings) of Q is at least c , while an R -group H may contain one or more c -cliques since the average similarity between a vertex of H and all vertices of H is at least c . In order to avoid working on an NP-Complete problem (the clique problem), the concept of R -group is introduced here for designing and studying approximation algorithms (Section 4).

2.2 Mathematical Model

Let S be a set of strings with the same length n , and let $\hat{A} : S \times S \rightarrow [0; 1]$ be similarity function between two strings.

The similarity relation of elements of S is modeled as a \hat{A} -weighted (complete) graph $K_{|S|}^{\hat{A}}$ with the vertex set S and $\hat{A}(x; y)$ as the \hat{A} -weight of the edge joining the vertices x and y (the similarity of the corresponding pair of strings x and y).

For a constant $c : 0 < c < 1$, the goal of motif finding is to find a subgraph H of $K_{jSj}^{\hat{A}}$ such that $\hat{A}(x;y) \geq c$ for every pair of vertices $x,y \in V(H)$. This optimization problem is actually the clique finding problem in general graph theory and discrete optimization areas, and is described in the following paragraph.

Let c be a positive number between 0 and 1 and let G^c be a subgraph of $K_{jSj}^{\hat{A}}$ with the same vertex set $S = V(K_{jSj}^{\hat{A}})$ and $E(G^c) = \{e \in E(K_{jSj}^{\hat{A}}) : \hat{A}(e) \geq c\}$. The graph G^c is called the c -truncated subgraph of $K_{jSj}^{\hat{A}}$.

The problem of finding a motif is equivalent to the problem of finding a clique of a given order m in the c -truncated graph G^c . Unfortunately, finding a clique of size $\geq m$ in graph is an NP-Complete problem [6]. Due to the hardness of the problem in this simple model, we would like to introduce an approximation approach that uses some information of the \hat{A} -weighted graph.

Let H be a subgraph of $K_{jSj}^{\hat{A}}$. Construct a new graph H^R from H by adding a special vertex $R(H)$ (called the representative of H). The new vertex does not correspond to a string; it statistically represents the distribution of the symbols of all elements of H . The Similarity function (\hat{A}) between the representative $R(H)$ and other vertices is defined in the previous subsection. The solution of the following problem is called an R -group of $K_{jSj}^{\hat{A}}$ and is a practical approximation of the clique (motif) problem.

Find a subgraph H of $K_{jSj}^{\hat{A}}$ such that $\hat{A}(R(H);x) \geq c$ for every $x \in V(H)$.

Though the simple model of c -truncated graph cannot lead us to an efficient algorithm for the motif problem, we may still occasionally use it as an auxiliary graph for some parts of the processing and discussion. However, our main attention is on the \hat{A} -weighted graph $K_{jSj}^{\hat{A}}$ and the representatives of some subgraphs.

3 Properties of the Model

Some properties about similarity estimation between vertices, c -cliques and R -groups are discussed in this section, which are theoretically necessary for the analysis of our model and algorithm.

3.1 Triangle Inequality

The triangle inequality is fundamental for any distance measure. Proposition 3.1 is a well-known fact for Hamming distance. Proposition 3.2, the counterpart of Proposition 3.1 for similarity, is an immediate corollary of Proposition 3.1.

However, the newly defined distance and similarity involved with representatives are not standard. Some inequalities for the estimation of similarities in our new model and some recursive operations are to be studied in this section.

Proposition 3.1 (Triangle inequality of Hamming Distance among three strings) If strings X , Y , and Z have the same length n and $d_H(X; Y) = c_1$ and $d_H(Y; Z) = c_2$, then $d_H(X; Z) \leq c_1 + c_2$, in which c_1 and c_2 are both constants.

Proposition 3.2 (Triangle inequality of similarity among three strings) If strings X , Y , and Z have the same length n and $\hat{A}(X; Y) = c_1$ and $\hat{A}(Y; Z) = c_2$, then $\hat{A}(X; Z) \leq \max(c_1 + c_2 - 1; 0)$, in which c_1 and c_2 are both constants.

Propositions 3.1 and 3.2 are to be generalized for not only strings but also representatives of subsets of strings.

Theorem 3.3 (Triangle inequality of similarity among two strings and one representative) Let H be a set of strings and $S_1; S_2$ be two strings (all of them have the same length n). If $\hat{A}(R(H); S_1) = c_1$ and $\hat{A}(R(H); S_2) = c_2$ ($0 \leq c_1; c_2 \leq 1$), then $\hat{A}(S_1; S_2) \leq \max(c_1 + c_2 - 1; 0)$.

Corollary 3.4 Let d be a real number, $0 \leq d \leq 1$. If H is a set of strings such that $\hat{A}(R(H); S) \leq (1 - d)$ for any $S \in H$, then $\hat{A}(S_i; S_j) \leq \max(1 - 2d; 0)$ for each $S_i; S_j \in H$.

This corollary can be restated as the following.

Corollary 3.5 Let d be a real number, $0 \leq d \leq 1$. If H is a set of strings such that $\hat{A}(R(H); S) \leq (1 - d)$ for each $S \in H$, then H is a c -clique, where $c = \max(1 - 2d; 0)$.

Theorem 3.6 (Triangle inequality of similarity among two representatives and one string) Let X and Y be two sets of strings and S be a string (all the strings here have the same length n). If $\hat{A}(R(X); S) = c_1$ and $\hat{A}(R(Y); S) = c_2$, then $\hat{A}(R(X); R(Y)) \leq \max(c_1 + c_2 - 1; 0)$.

Theorem 3.7 (Another triangle inequality of similarity among two representatives and one string) Let X and Y be two sets of strings and S be a string (all the strings here have the same length n). If $\hat{A}(R(X); S) = c_1$ and $\hat{A}(R(X); R(Y)) = c_2$, then $\hat{A}(S; R(Y)) \leq \max(c_1 + c_2 - 1; 0)$.

Theorem 3.8 (Triangle inequality of similarity among three representatives) Let X , Y and Z be three sets of strings with the same length n . If $\hat{A}(R(X); R(Y)) = c_1$ and $\hat{A}(R(Y); R(Z)) = c_2$ ($0 \leq c_1; c_2 \leq 1$), then

$$\hat{A}(R(X); R(Z)) \leq \max(c_1 + c_2 - 1; 0):$$

3.2 Operations on Cliques and R-groups

Recursive operations (such as, insertions and deletions) are key elements of an algorithm.

In this subsection, changes of similarities after operations are to be estimated in several inequalities.

Theorem 3.9 (Insertion) Let S be a set of strings with the same length n ($|S_j| = k$). Let T be a string of length n that is not in S , and S be a string in S . If $\hat{A}(R(S); T) = c_1$, $\hat{A}(R(S); S) = c_2$ and $\hat{A}(S; T) = c_3$, then

$$\hat{A}(R(S \cup \{T\}); T) = c_1 + (1 - c_1)(k + 1)$$

and

$$\hat{A}(R(S \cup \{T\}); S) = c_2 + (c_3 - c_2)(k + 1):$$

Corollary 3.10 Let S be a set of strings with the same length n ($|S_j| = k$). Let T be a string of length n that is not in S , and S be a string in S . If $\hat{A}(R(S); S) \geq c$ and $\hat{A}(T; S) \geq c$, then

$$\hat{A}(R(S \cup \{T\}); S) \geq c:$$

By Theorem 3.9, we note that if $c_2 \geq c_3$, then $\hat{A}(R(S \cup \{T\}); S) \geq \hat{A}(R(S); S)$:

Theorem 3.11 (Deletion) Let S be a set of strings with the same length n ($|S_j| = k$). If $\hat{A}(R(S); S) = c_1$, $\hat{A}(R(S); T) = c_2$ and $\hat{A}(S; T) = c_3$ (where S and T are two strings in S), then $\hat{A}(R(S \setminus \{S\}); T) = c_2 + \frac{c_2 - c_3}{k - 1}$.

Theorem 3.12 (Deletion) Let S be a set of strings with the same length n ($|S_j| = k$) and let $S \in S$ with $\hat{A}(R(S); S) = c$, then

$$\hat{A}(R(S); R(S \setminus \{S\})) = 1 - \frac{1 - c}{k - 1}:$$

Corollary 3.13 S is an R-group with similarity c ($|S_j| = k$) if and only if $\hat{A}(R(S); R(S \setminus \{S\})) \geq 1 - \frac{1 - c}{k - 1}$ for any $S \in S$.

4 Algorithm

4.1 Basic Algorithm

Our goal of the following algorithm is to find a collection of R-groups such that if a vertex v is contained in a clique of order m , it is contained in some of the output R-groups.

Algorithm 4.1

Input: S : a set of k strings with length n , a constant integer m and a real number $c : 0 < c < 1$ (the similarity lower bound for R-groups).

step 1: Construct the \hat{A} -weighted (complete) graph $K_{jS_j}^{\hat{A}}$ with the vertex set S and $\hat{A}(x; y)$ as the \hat{A} -weight of the edge joining any vertices x and y (the similarity of the corresponding pair of strings x and y). Build the related c -truncated graph K^c . (Complexity $\circ (k^2n)$.)

step 2: Find the vertex u with the maximum degree in K^c . (This can be done in conjunction with step 1.)

step 3: Let G^0 be the subgraph of the c -truncated graph K^c induced by the closed neighborhood $N[u]$ of u . Let G be the subgraph of the \hat{A} -weighted graph $K_{jS_j}^{\hat{A}}$ induced by same vertex set as that of G^0 . (Complexity is $\circ (k^2)$.)

step 4: Update G and G^0 by removing all vertices with degrees less than $m - 1$ in G^0 . (Complexity $\circ (k^2)$.)

step 5: Let $jG^0j = k^0$. (If $k^0 < m$, then output an empty set and stop.) Find the vertex s in G^0 such that $\hat{A}(R(G^0); s)$ is minimum. Check if $\hat{A}(R(G); R(G \text{ n fsg})) \leq 1 - \frac{1-c}{k^0-1}$. If "yes", then go to step 7; otherwise, go to step 6. (Complexity is $\circ (k^2n)$.)

step 6: Let M be the vertex subset of G^0 such that every element v of M satisfies $\hat{A}(R(G); v) < c$. Let $d(v)$ be the degree of v in G^0 . Select a vertex s^0 from M such that $d(s^0) \notin \hat{A}(R(G^0); s^0)$ is minimum. Update G and G^0 by removing s^0 . Go to step 5. (Complexity is $\circ (k^2n)$.)

step 7: Output an R-group. (Total complexity is $\circ (k^3n)$.)

Repeating the algorithm. The algorithm above is to find an R-group containing a specific large degree vertex. After we have found an R-group, we can find others by starting at another vertex u with large degree in K^c , which is not contained in any output and not been used in step 2 yet. Then iterate step 3 to step 7.

4.2 Modifications for Various Applications

In real applications, biologists and other scientists may have some motif finding problems with various requirements. Thus, modifications of the models and the algorithm are needed for those related problems. Discussions in the following

cases briefly describe the modifications for some typical problems.

Case 1. The input is a set X of sequences and an integer n . The desired output is a motif H consisting of substrings (of the same length n) of some member of X . Here, the number of substrings contributed to H from each member of X is not restricted.

The models and algorithm presented in the previous sections can be applied directly to this type of problem: one can simply let S be the set of all substrings (of the same length n) of some members of X .

Case 2. The input is a set X of sequences and an integer n . The desired output is a motif H consisting of substrings (of the same length n) of some member of X . Here, the number of substrings contributed to H from each member of X is precisely one.

For this problem, the λ -weighted graph $K_{jS_j}^\lambda$ should be modified as a λ -weighted multipartite complete graph in which the i -th part of the vertex partition is the set of all substrings of the i -th member of X . Since there is no "similarity" defined for substrings from the same sequence, with a few further restriction in Algorithm 4.1, each output consists of precisely one substring from each member of X . This is an outline of modification, and the details are omitted here.

Problems of Case 2 are more complicated than problems of Case 1 if one applies the algorithm and its idea in the previous sections. Because of the totally different approaches, it is not surprising that the Gibbs sampling algorithm [9] has an easier processing for problems of Case 2 than problems of Case 1 (see Chapter 10 of [12]).

References

- [1] Akutsu, T.: Hardness result on gapless local multiple sequence alignment, Technical Report 98-MPS-24-2, Information Processing Society of Japan, 1998.
- [2] Akutsu, T., Arimura, H., and Shimozone. S.: On approximation algorithms for local multiple alignment, RECOMB00: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, Tokyo, Japan, April 2000.
- [3] Bailey, T. L. and Elkan, C.: Unsupervised learning of multiple motifs in biopolymers using expectation maximization, *Machine Learning*, 21 (1-2):51-80. October 1995.

- [4] Bondy, J. A. and Murty, U. S. R.: Graph Theory with Applications, Macmillan, London, 1976.
- [5] Garey, M. R. and Johnson, D. S.: Computers and Intractability, W. H. Freeman and Company, San Francisco, 1979.
- [6] Gibbons, A.: Algorithmic Graph Theory, Cambridge University Press, Cambridge, MA, 1985.
- [7] Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge Univ. Press., Cambridge, MA, Jan. 15, 1997.
- [8] Hertz, G. Z. and Stormo, G. D.: Identifying DNA and protein patterns with statistically significant alignments of multiple sequences, *Bioinformatics*, 15 (7/8):563-577, July/August 1999.
- [9] Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. and Wootton, J. C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, *Science*, 262:208-214, 8 October 1993.
- [10] Pearson, W. R.: Rapid and Sensitive Sequence Comparison with FASTP and FASTA, *Methods in Enzymology*, 183:63-98, 1990.
- [11] Smith, T. F. and Waterman, M. S.: Identification of common molecular subsequences, *Journal of Molecular Biology*, 147 (1):195-197, March 1981.
- [12] Tompa, M.: "Lecture Notes on Biological Sequence Analysis", Department of Computer Science and Engineering, University of Washington, 2000, Chapter 10.
- [13] West, D. B.: "Introduction to Graph Theory", Prentice Hall, Upper Saddle River, NJ., 1996.