BIOINFORMATICS ORIGINAL PAPER

Vol. 22 no. 1 2006, pages 21–28
doi:10.1093/bioinformatics/bti745

Sequence analysis

A generic motif discovery algorithm for sequential data

Kyle L. Jensen[1], Mark P. Styczynski[1], Isidore Rigoutsos[1,2] and Gregory N. Stephanopoulos[1,*]

[1]Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and [2]IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Received on August 4, 2005; revised on October 14, 2005; accepted on October 24, 2005
Advance Access publication October 27, 2005
Associate Editor: Keith A Crandall

ABSTRACT

Motivation: Motif discovery in sequential data is a problem of great interest and with many applications. However, previous methods have been unable to combine exhaustive search with complex motif representations and are each typically only applicable to a certain class of problems.

Results: Here we present a generic motif discovery algorithm (Gemoda) for sequential data. Gemoda can be applied to any dataset with a sequential character, including both categorical and real-valued data. As we show, Gemoda deterministically discovers motifs that are maximal in composition and length. As well, the algorithm allows any choice of similarity metric for finding motifs. Finally, Gemoda's output motifs are representation-agnostic: they can be represented using regular expressions, position weight matrices or any number of other models for any type of sequential data. We demonstrate a number of applications of the algorithm, including the discovery of motifs in amino acids sequences, a new solution to the (l,d)-motif problem in DNA sequences and the discovery of conserved protein substructures.

Availability: Gemoda is freely available at http://web.mit.edu/bamel/gemoda

Contact: gregstep@mit.edu

Supplementary Information: Available at http://web.mit.edu/bamel/gemoda

INTRODUCTION

Motif discovery encompasses a wide variety of methods used to find recurrent trends in data. In bioinformatics, the two predominant applications of motif discovery are sequence analysis and microarray data analysis. Less common applications include discovering structural motifs in proteins and RNA (Holm *et al*., 1992; Murthy and Rose, 2003).

Motif discovery in sequence analysis typically involves the discovery of binding sites, conserved domains, or otherwise discriminatory subsequences. There are many publicly available tools, each of which is quite adept at addressing a specific subclass of motif discovery problems. Some of the commonly used tools for motif discovery in nucleotide and amino acid sequences include MEME (Bailey and Elkan, 1994), Gibbs sampling (Lawrence *et al*., 1993), Consensus (Hertz and Stormo, 1999), Block Maker (Henikoff *et al*., 1995), Pratt (Jonassen *et al*., 1995) and Teiresias (Rigoutsos and Floratos, 1998). Newer, less-widely used tools include Projection (Buhler and Tompa, 2001), MultiProfiler (Keich and Pevzner, 2002), MITRA (Eskin and Pevzner, 2002) and ProfileBranching (Price *et al*., 2003). This list is not intended to be exhaustive; however, it is indicative of the wealth of options available for solving such problems.

All of the existing motif discovery tools for nucleotide and amino acid sequences can be classified on a spectrum ranging from exhaustive tools using simple motif representations to non-exhaustive tools using more complex representations. The majority of the tools can be found at the extreme ends of the spectrum, with tools that exhaustively enumerate regular expressions (or single consensus sequences) at one end and probabilistic tools, based on position weight matrices (PWMs), at the other. This partitioning of tools is due to a computational trade-off: more descriptive motif representations such as PWMs frequently make exhaustive searches computationally infeasible.

Depending on the task at hand, a specific type of motif discovery tool may be more useful than others. For example, the PWM-based tools excel at finding *cis*-regulatory binding elements (Tompa *et al*., 2005), whereas the regular expression-based tools are well-suited to finding conserved domains in large protein families (Rigoutsos *et al*., 1999). Generally, it can be difficult to know a priori which motif discovery tool will be most appropriate.

ALGORITHM

Gemoda was designed to meet the demand for complex motif representations, like PWMs, while still being exhaustive. The philosophical underpinnings of the Gemoda algorithm can be traced back to Teiresias (Rigoutsos and Floratos, 1998); Winnower (Pevzner and Sze, 2000); the algorithm by Mancheron and Rusu (2003) and a variety of algorithms for association mining (Zaki, 2000; Zaki and Ogihara, 1998). In particular, Gemoda shares some of its logical steps with the Teiresias algorithm while incorporating a more flexible definition of 'similarity' and allowing motif representations other than regular expressions.

Gemoda's design goals can be summarized as follows: exhaustive discovery of all maximal motifs in a way that allows flexibility in motif representation, incorporation of a variety of similarity metrics and the ability to handle diverse sequential data types. Each major point can be explained as follows:

- Exhaustive discovery: Gemoda's combinatorial nature provides an algorithmic guarantee that all motifs meeting certain criteria are deterministically discovered.

*To whom correspondence should be addressed.

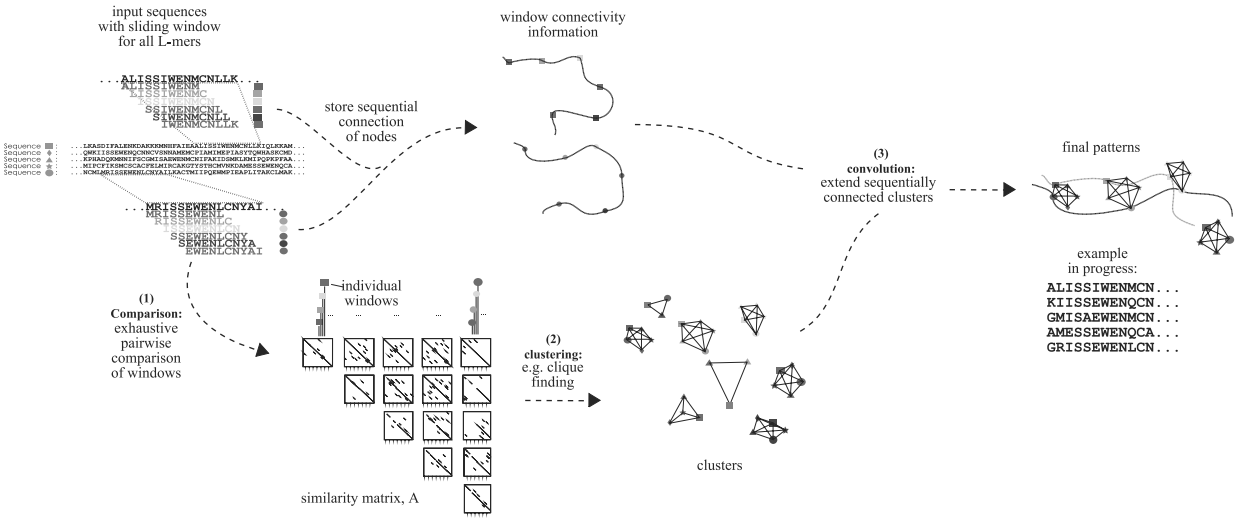

Fig. 1. A sketch showing the flow of the Gemoda algorithm for an example input set of protein sequences. The various colors in the input sequences are used to indicate the sequential ordering of the L-residue windows. The various shapes are used to indicate a particular window's sequence of origin. (1) In the comparison stage, each window is compared with each other window on a pair-wise basis. Here we show the similarity matrix, A, where the values in the matrix have been thresholded. Those pairs of windows in A that have a similarity score above the threshold are colored black. Note that the graph looks very similar to a standard dot plot. (2) In the clustering phase, groups of windows are clustered together. Here, we show the clusters as cliques, or maximal fully connected subgraphs in the thresholded matrix A. (3) Finally, these clustered are 'stitched' together in the convolution phase using the sequential ordering of the windows to reveal the maximal motifs. A similar process applies for any kind of sequential data analyzed by Gemoda.

- Maximal motifs: Gemoda returns only motifs that are maximal in both length and composition with respect to the similarity and clustering functions.
- Motif representation: The motifs discovered by Gemoda are reported as short multiple sequence alignments (in the case of motif discovery in nucleotide and amino acid sequences) and can be modeled using regular expressions, PWMs/PSSMs, Markov models or any other representation.
- Similarity metrics: Any criterion, ranging from sequence alignment scores to geometric functions, may be used to compare sequences.
- Sequential data types: The nature of Gemoda's computations is not unique to any specific type of data, and thus can be used on any data with a sequential character—i.e. data in which there is a natural left-to-right order, such as a sequence of nucleotides or amino acids. In the most general sense, sequential data also include real-valued series data, such as a stock price or the ordered (x, y, z) triplets of an alpha-carbon trace in a protein structure.

The algorithm has three distinct phases: comparison, clustering and convolution. During the comparison phase, short overlapping windows in the dataset are compared. During clustering, these windows are grouped together to form elementary motifs. Finally, during convolution, these motifs are 'stitched' together to form maximal motifs (Fig. 1). In the following sections, we give some brief definitions and nomenclature, then describe each of the algorithm's three phases in detail. Finally, we illustrate a few applications of Gemoda.

Preliminary definitions and nomenclature

The input to Gemoda is a set of sequences of data points $S = \{s_1, s_2, \ldots, s_n\}$, where sequence s_i has length W_i. So, e.g. the j-th member of the i-th sequence is denoted by $s_{i,j}$. Each $s_{i,j}$ is a primitive, or atomic unit, for the data that are being analyzed. For time-series data, $s_{i,\,j}$ may be a point sampled from $\mathbb{R}^K$ (with K arbitrary), whereas for a DNA sequence it would be one of the characters {A, T, G, C}.

Typically, one seeks motifs of a minimal, domain-dependent length. We denote this minimum length by L and we define a matrix A of size $N \times N$, where $N = \sum_{i=1}^{n} (W_i - L + 1)$. That is, A is a matrix with one row and one column for each window of size L in our entire sequence set. For example, the 10th window of size L in the 5th sequence would be expressed as $s_{5,10:10+L-1}$, where '$10 : 10 + L - 1$' denotes 'position 10 through position $10 + L - 1$, inclusive.' To keep track of which window corresponds to which index in A, we define the one-to-one function $\mathscr{M}(s_{i,j:j+L-1}) \mapsto q \in [1, N]$. (For simplicity, we define $(s_{i,j} + 1)$ to be $s_{i,j+1}$, unless $s_{i,j+1}$ does not exist, in which case $(s_{i,j} + 1)$ is undefined.) Similarly, $\mathscr{M}^{-1}(q) \mapsto (s_{i,j:j+L-1})$ such that $i \in [1, n]$ and $j \in [1, W_i - L + 1]$.

We also define a similarity function $\mathscr{S}(s_{i,j:j+L-1}, s_{q,z:z+L-1})$, that takes as arguments two arbitrary windows and returns a real-valued number indicating the level of similarity between the two windows. In the most simple case, $\mathscr{S}$ may use the identity matrix to count how many DNA bases two windows have in common; for real-valued data, the function may return the sum-of-squares error between two windows or any other measure of similarity.

We define a motif p as a data structure with two features: a width $\mathscr{W}(p)$ and a list of locations in the data where the motif occurs, $\mathscr{L}(p)$. A motif has the property that the locations in $\mathscr{L}(p)$ meet some predefined clustering requirements (discussed below) based on the similarity function $\mathscr{S}$ for each window of length L within the motif. The support of a motif is equal to the number of its occurrences (or 'embeddings'), $|\mathscr{L}(p)|$.

We say a maximal motif is a motif which has the following properties:

(1) The motif's width cannot be extended in either direction (left or right) without producing a motif with fewer embeddings (i.e. without $|\mathscr{L}(p)|$ decreasing) and

(2) The motif is not missing any instances, i.e. $\mathscr{L}(p)$ includes the locations of all instances of the motif.

These two criteria can be summarized qualitatively by stating that a maximal motif is not 'missing' any locations and is as wide as possible, and thus it is as specific and sensitive as possible.

Given these explanations and definitions, we can now detail the computations involved in each phase of the Gemoda algorithm. A simple natural-language example illustrating how each phase proceeds is included in the Supplementary materials.

Comparison phase

In the comparison phase of the Gemoda algorithm, the sequences are divided into overlapping windows of size L which are then compared with each other in a pairwise manner to produce a similarity matrix, A (Fig. 1). Formally, $A_{i,j}$ is equal to $\mathscr{S}(\mathscr{M}^{-1}(i),\mathscr{M}^{-1}(j)) = \mathscr{S}(s_{i,j:j+L-1}, s_{q,z:z+L-1})$. A is then, quite simply, a similarity matrix for all N windows based on the similarity function $\mathscr{S}$. In most cases, $\mathscr{S}$ is commutative (and the A matrix is symmetric); however, this is not a requirement.

Clustering phase

The purpose of the clustering phase is to use the similarity matrix A to group similar windows in clusters. These clusters will become 'elementary motifs' from which the final, maximal motifs will be constructed.

We define a clustering function $\mathscr{C}(A) = c^L = \{c_1^L, c_2^L, \ldots, c_Z^L\}$ where each c_i^L is a set of indices in A and $c_i^L[q]$ is the q-th member of c_i^L. Note that $\mathscr{C}$ can be any function; common clustering functions include hierarchical clustering, k-nearest-neighbors clustering and many others. We call each c_i^L an 'elementary motif' of length L. We note that a clustering function may assign each node (window) to one or more groups. In the latter case, each c_i^L may have a non-null intersection with any c_j^L.

Convolution phase

The purpose of this phase is to 'stitch together' the elementary motifs to generate the final, maximal motifs (Rigoutsos and Floratos, 1998). For the purposes of Gemoda (and consistent with the above concept of convolution), we say that a motif h of width $\mathscr{W}(h) > L$ meets the similarity criterion if for each window of length L completely within the motif, all instances participate in a cluster together based on $\mathscr{S}$ and $\mathscr{C}$. In this manner, we can piece together longer continuous motifs from smaller motifs that all meet the similarity criterion over windows of length L.

Next we define the 'directed intersection' of two elementary motifs, $c_i^L \curvearrowright c_j^L = c_r^{L+1}$, where c_r^{L+1} is the set of those indices q in c_i^L such that $\mathscr{M}(\mathscr{M}^{-1}(c_i^L[q])+1)$ is in c_j^L. That is, c_r^{L+1} is the set of indices in c_i^L that are located, in the sequences S, one position earlier than the indices in c_j^L. c_r^{L+1} is then a motif of length $L+1$.

We define the operation '$\sqsubset$' as follows: $c_i^L \curvearrowright c_j^L \sqsubset c^{L+1}$ is true if the set of indices $c_i^L \curvearrowright c_j^L$ is a subset or a superset of the indices in any member of c^{L+1}. This operation compares a convolved motif of length $L+1$ with all previously-convolved motifs of length $L+1$ to identify significant overlap: if the list of locations in the proposed motif is a superset or subset of the list for any other motif, the result of this operation is true. With this step, Gemoda can identify and eliminate redundant and non-maximal motifs. If $c_i^L \curvearrowright c_j^L \sqsubset c^{L+1}$, then all super- or subsets of the proposed convolved motifs are removed from c^{L+1}; these windows are then taken together with the proposed motif, and the union of those sets of windows is returned to c^{L+1}.

Our objective is to find all the maximal motifs in the sequence set using the elementary patterns. We do this by performing $c_i^k \curvearrowright c_j^k$ for all i and j at each length $k \geq L$ until c^k is empty ($|c^k| = 0$). We then define the set of maximal motifs comprising c^k for all k as P, the final set of motifs that are returned to the user. This simple induction scheme guarantees that all (and only) the maximal motifs are in P given appropriate clustering functions (see Supplementary materials).

Implementation

Choice of clustering function Gemoda can use any clustering function; however, as the size of the input sequence set increases, storing the matrix A can become practically difficult. In these cases, it can be easier to store true/false values in A, where the value is true if the similarity score between two windows is better than a user-defined threshold g. The matrix A can then be viewed as an unweighted, undirected graph with a vertex for each window and edges between those nodes with pairwise similarity scores better than g (Figs. 1 and 2). When constructed as such, we have found that clustering functions based on finding either cliques[1] or connected components (maximal disjoint subgraphs) can be effective for motif discovery in diverse applications.

In the case where the clustering function $\mathscr{C}(A)$ is chosen such that each c_i^L is a clique in the g-thresholded A matrix, the Gemoda algorithm has a guarantee of compositional and length maximality, relative to the threshold g. That is, Gemoda will discover all motifs where each pair of instances has a similarity score better than g over every window of size L, there are no 'missing' instances having this property and the motif cannot be extended either to the left or right (see inductive proof in the Supplementary material).

Clique enumeration is NP-complete (Garey and Johnson, 1979; Tomita *et al*., 1989); however, in practice this complexity is usually not an issue because the density (the ratio of the number of edges to the number of vertices) of graphs is usually low for datasets of nucleotide or amino acid sequences (with reasonable choice of g).

In the case where the clustering function $\mathscr{C}(A)$ is chosen such that each c_i^L is a maximal disjoint subgraph in the g-thresholded A matrix (i.e. c^L represents the connected components of A), the computational complexity for the clustering phase is significantly less than for clique-based clustering. As well, in the case where Gemoda is applied to nucleotide and amino acid sequences, the motifs from this connected components method may be more intuitive than motifs found using clique-based clustering.

[1]We define a clique as a maximal, fully connected subgraph. It may be alternatively defined without the requirement for maximality, thus making the clusters we discuss 'maximal cliques'. We use the former definition for the sake of brevity and clarity when discussing the maximality of extending motifs.

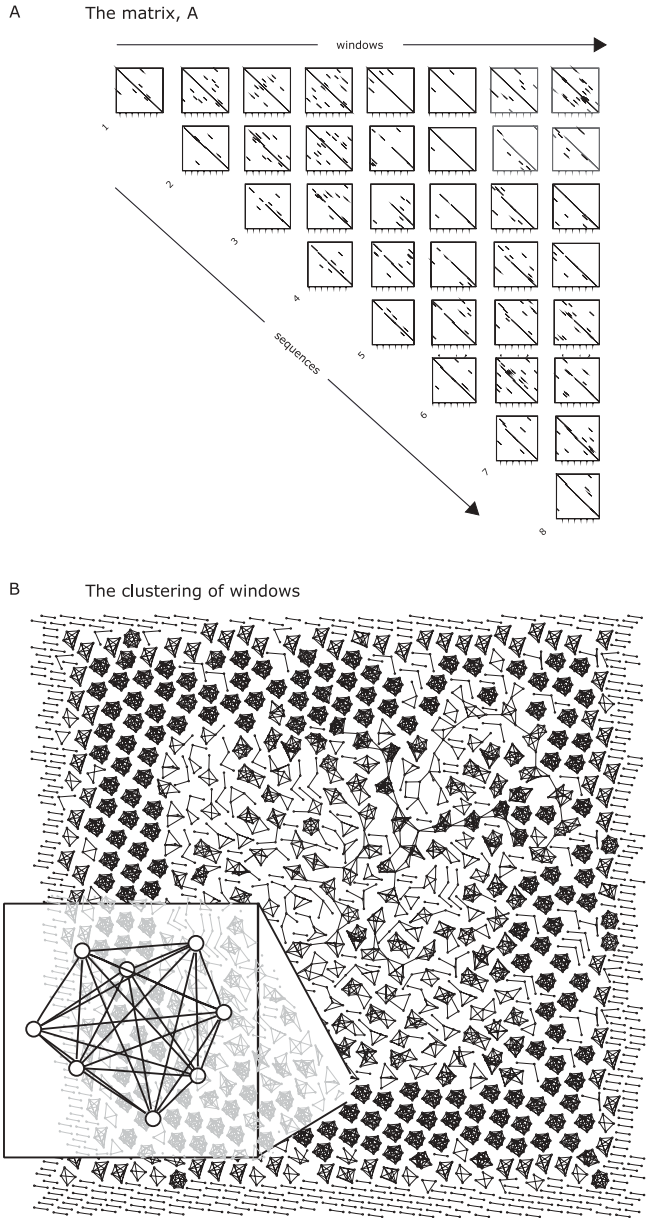

Fig. 2. The similarity graph for the 3.1.7.2 enzyme example. (**A**) is the similarity matrix *A*, which contains one row and column for each window of 50 residues in the set of input sequences. Entries in the matrix have been thresholded such that pairs of windows that can be aligned with a bit-score greater than 20 are given a black dot and all others are white, producing the familiar dot–plot appearance of the matrix. (**B**) is a graph representation of *A*. Each vertex represents a window, and two vertices are connected with an edge if they have a black dot in the top image. The breakout shows a clique of size eight, which represents a set of windows that participate in the motif shown in Figure 3. In general, as the bit-score threshold is lowered, the number of edges in the graph increases, making the clustering stage more computationally intensive. When using clique-based clustering with too small of a threshold, computational expense may make the problem infeasible. At these thresholds the 'signal' cannot be distinguished from the 'noise.' However, with the parameters used in this example, the clustering phase is quite easy, which is intuitive given the number of disjoint subgraphs shown in the bottom image.

The space and time usage of this implementation is not unreasonable. In most cases, memory usage is not a limiting factor. For instance, the peak memory usage for a large sequence set containing 65 000 characters is 1 GB, within the reach of many personal computers. Furthermore, the upcoming examples given in this work can all be done in reasonable times. The amino acid sequence example and the protein structure example take at most tens of seconds on an average desktop PC, while the hardest of the DNA sequence examples takes 2 h. These times are more than reasonable given the exhaustive guarantees provided by the algorithm.

Estimation of motif significance The absolute significance of motifs depends strongly on the choice of the similarity metric and clustering function and is difficult to derive a priori. However, for a specific pair of similarity metric and clustering function, the relative significance can be easy to calculate. For the clique-based clustering function described above, the relative significance can be estimated solely from the matrix *A* using a bootstrapping method. (A description of this calculation is included in the Supplementary materials.) Such significance calculations are equally valid for many different motif discovery problems (e.g. nucleotide sequences or protein structures) because the calculation method uses only the matrix *A*: it is data-type agnostic.

Summary of user-supplied parameters The input to Gemoda is a set of sequences (categorical or real-valued), a window length, a similarity function and a clustering function. Various clustering functions may require other parameters. For example, the clique-finding and connected components clustering algorithms discussed above require threshold parameter *g* and, optionally, a minimal support parameter *k*. Other parameters can be easily incorporated into various clustering functions, such as a 'unique support' parameter *p* that limits returned motifs to those that occur in at least *p* different sequences.

Availability We have written open source programs implementing the Gemoda algorithm that are publicly available at the following URL: http://web.mit.edu/bamel/gemoda. The software includes a number of 'helper' applications for interoperability with common bioinformatics tools. For example, applications are included that allow users to model Gemoda's output motifs (in the case of nucleotide or amino acid sequences) as PSSMs—using the pftools package available via the Prosite database (Hofmann *et al*., 1999)—or as hidden Markov models, using the popular HMMer software (Eddy, 1998).

The implementation is distributed in two variants, each with a different comparison stage of the algorithm. The gemoda–s variant is for motif discovery in FastA-formatted text strings, typically nucleotide or amino acid sequences. The gemoda–r variant is used for motif discovery in sets of multidimensional, real-valued sequences. The gemoda–s variant is distributed with a number of similarity functions based on various nucleotide and amino acid substitution matrices. The gemoda–r variant is distributed with similarity functions based on the root mean square deviation, with options for optimal translation and rotation.

APPLICATION

In this section, we demonstrate Gemoda's capability by presenting several sample applications. Specifically, we address motif

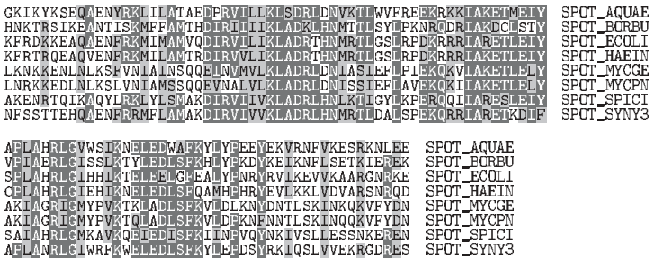

Fig. 3. The RelA_SpoT motif detected in the 3.1.7.2 enzyme sequences.

discovery in amino acid sequences, in nucleotide sequences and in protein structures.

As discussed previously, the clustering and convolution stages of the Gemoda algorithm are generic—they are independent of the nature of the input data. However, the comparison stage is data-specific. In what follows, we discuss how the comparison stage is changed for each kind of data and outline the types of results Gemoda is capable of finding.

Motif discovery in amino acid sequences

To use Gemoda to find motifs in amino acid sequences, the comparison stage needs to reflect the notion of 'similarity' for amino acid sequences. Specifically, we choose a window comparison function $\mathscr{S}$ that returns a sequence alignment score, such as the bit-score from an amino acid scoring matrix [e.g. the popular Blosum matrices (Henikoff and Henikoff, 1992)].

Here, we demonstrate how Gemoda can be used for motif discovery in amino acid sequences by 'discovering' known protein domains in the (ppGpp)ase family of enzymes. These eight enzymes catalyze the hydrolysis of guanosine 3′, 5′-bis(diphosphate) to guanosine 5′-diphosphate (GDP) and are classified by the Enzyme Commission (EC) number 3.1.7.2 (Bairoch, 2000).

We used Gemoda to identify motifs in these eight (ppGpp)ase enzymes using the Blosum–62 scoring matrix as the basis of our similarity function $\mathscr{S}$ and the clique-based clustering function described previously. Specifically, we sought motifs that occurred in all 8 sequences, were at least 50 residues long and had a pairwise bit-score of at least 50 bits over a window of 50 residues.

With these parameters, Gemoda discovers 4 motifs in this set of 8 sequences; the longest motif, with a length of 103 amino acids, is shown in Figure 3 as an alignment of the regions that correspond to instances of this motif (Fig. 2). A comparison with the known protein domains in the NCBI Conserved Domain Database (Version 2.02) (Marchler-Bauer *et al*., 2003) reveals that this motif captures the RelA_SpoT domain (CDD PSSM–id 15904).

The remaining three motifs are not present in the CDD database. However, further inspection using the tools available from the PFAM database (Bateman *et al*., 2004) revealed that they composed the left, middle and right regions of the HD domain (Aravind and Koonin, 1998). In the SpoT enzymes, this domain has a number of insertions and deletions that give rise to gaps such that Gemoda identified and reported individually the left, middle and right regions of conservation of the HD domain.

In this example, the Blosum-62 matrix was chosen as the similarity metric because it is optimized for detecting distant homologs. The Gemoda input parameters $L = 50$ and $g = 50$ were chosen to enforce a one-bit-per-base score, which should rise above random 'noise' since, by design, the expected bit-score for two aligned amino acids is negative for the Blosum set of scoring matrices.

In order to test the sensitivity of these results to noise, we conducted an experiment to determine the degree to which these (ppGpp)ase motifs could be found if obscured by noise caused by adding random spurious sequences to the eight enzyme sequences. We found that, with the Gemoda input parameters described above and using random sequences selected from Swiss-Prot (Release 45.0) (Bairoch and Apweiler, 2000), the target motifs could be detected in an 8-fold majority of spurious sequences.

Motif discovery in nucleotide sequences

The discovery of motifs in nucleotide sequences is most commonly used in the search for *cis*-regulatory elements. The 'Motif Challenge Problem,' or the (l, d)-motif problem (Pevzner and Sze, 2000), is an abstraction of the *cis*-regulatory element discovery problem.

The original (l, d)-motif problem can be paraphrased as follows:

> Within a set of random DNA sequences with i.i.d. nucleotides, a parent motif of length l is embedded in each sequence in a random location. Each time the motif is embedded, it is mutated in d locations. The (l, d)-motif problem is to recover the locations of the embeddings, knowing only the parameters l and d and that each sequence contains exactly one instance of the motif.

To a certain extent, this is a somewhat reasonable abstraction of the *cis*-regulatory element discovery problem. It is also a problem in which false positive motifs are not expected to occur by chance: the occurrence of a motif with an instance of d or less mutations in each of the 20 sequences has a probability of approximately 10^{-15} for $l = 15$ and $d = 4$ (Buhler and Tompa, 2001). However, the probability is 0.057 that any two windows of length 15 may be 4 mutations from a common ancestor. In a set of 20 sequences each of length 600, one would then expect any given window to be 'similar' to 663 other windows purely by chance. With such significant noise obscuring the smaller, easily-identifiable signal, this is a difficult problem that, as Pevzner and Sze (2000) pointed out, commonly used tools are incapable of solving accurately.

Gemoda can provide a direct solution to this problem, using clique-based clustering and a comparison function based on the identity matrix. The selection of g is simple, as any two motifs with d mutations in l positions must have $l - 2d$ bases in common. The only additional step necessary is to verify that each of the motif instances identified by Gemoda could have the same ancestor, a simple task. We have previously reported (Styczynski *et al*., 2004) that a dataset used by Pevzner and Sze (2001) in their initial presentation of the challenge problem in fact had an instance of the parent motif that occurred completely by chance and had gone otherwise undetected. With Gemoda, we can easily identify this instance without any additional work or manipulation. The sequence logo for the planted motif from Pevzner and Sze's initial dataset is shown in Figure 4; the consensus sequence is GGCTTTGTAGCTAAC. The 'accidental' instance of the embedded motif that can be identified using Gemoda is GGATTGATAGCTAAG.

Clearly, Gemoda was not originally designed to address the (l, d)-motif problem and, consequently, it does not exploit all of the characteristics of the problem to solve it in the fastest possible way. However, it does provide a direct, exhaustive solution to the problem that identifies otherwise undetectable results.

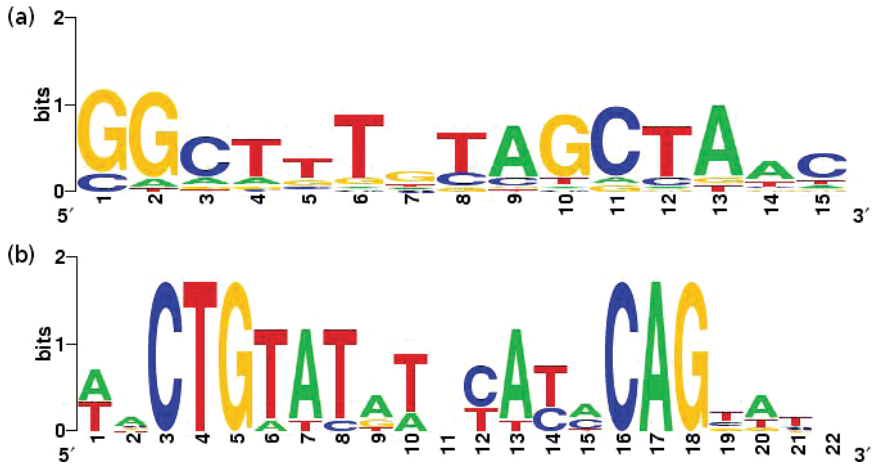

Fig. 4. The sequence logo for (A) the motif implanted in each sequence for the (l, d)-motif problem and (B) the LexA binding site motif generated from the highest-scoring motif returned by Gemoda. Logos created using WebLogo (Crooks *et al*., 2004).

Identifying natural cis–regulatory elements For some regulons in *Escherichia coli* with mild to strong consensus sequences, Gemoda returns results that are similar to or improve upon the results from commonly used motif discovery tools. For instance, using the set of upstream regions (400 bp upstream and 50 bp downstream of the translation start site) for the 9 operons believed to be regulated by LexA (Salgado *et al*., 2004), Gemoda's top-scoring motif was used to generate the sequence logo found in Figure 4. This motif closely matches the literature PWM for the LexA binding site and represents 80% of the literature-found binding sites with no false positives. Of course, the difficulty of DNA motif discovery problems varies greatly, and this is only one straightforward example of such problems.

The parameters used for this search were $L = 20$, $g = 10$ and $k = 6$ with the identity matrix scoring scheme and clique-based clustering described above. The length was selected based on the knowledge that the DNA-binding domain of LexA is a helix–turn–helix variant, and so it was likely to be a relatively long motif. The similarity threshold was chosen as one-half of L, which we know from the (l, d)-motif problem ought to be approximately sufficient to prevent the graph from being too dense (and thus expensive to cluster). The support threshold was chosen to be about two-thirds the total number of sequences, allowing for some noise in the data. Of course, the judicious selection of parameters is an outstanding problem in binding site discovery. It is worth noting that most of these selections were simple or intuitive and that there was some tolerance in the results for slight perturbations in parameters.

Motif discovery in protein structures

The detection of three-dimensional (3D) motifs in sets of protein structures is another problem type that Gemoda can address. Often, homologs that are related through a distant lineage show little to no sequence similarity, particularly at the nucleotide level (Eidhammer *et al*., 2000). However, these homologs frequently show conserved tertiary structures (Dietmann and Holm, 2001), making motif discovery in protein structures often revealing in situations where there appears to be no similarity at a sequence level.

There are a number of well-developed tools for the pair-wise comparison of protein structures or the comparison of a single protein structure to precomputed structural motifs; these have been reviewed elsewhere (Eidhammer *et al*., 2000). Some of the more popular tools include SSAP (Orengo and Taylor, 1996), VAST (Madej *et al*., 1995), Dali (Holm and Sander, 1993) and Mammoth (Ortiz *et al*., 2002). The Gemoda algorithm, when used for structural motif discovery, is most similar to the Sarf algorithm (Alexandrow, 1996; Alexandrov and Fischer, 1996) and, to a lesser degree, algorithms by Hunter and Subramaniam (2003) and Jonassen *et al*. (2002). Conceptually, Gemoda could be thought of as a hybrid of the Sarf and Teiresias algorithms, combining 3D elementary motif discovery with convolution. To the best of our knowledge, Gemoda is the only tool that can compare an arbitrary number of protein structures simultaneously and produce an exhaustive set of maximal motifs.

To discover motifs in protein structures, Gemoda compares L-residue windows of the proteins' alpha-carbon trace using the minimized RMSD similarity metric (one of many possible metrics for comparing protein substructures (Kolodny *et al*., 2005)). Here we use 'minimized' to indicate that the protein structures are optimally superimposed via rigid–body rotation and translation (Horn, 1987; Arun *et al*., 1987); occasionally this term is implicit. Using the clique-finding clustering algorithm, Gemoda finds motifs that are sets of alpha-carbon traces (in a set of protein structures) that can be super-imposed with an RMSD less than g Å over each window of L-residues on a pair-wise basis. Similar to the amino acid and nucleotide applications of Gemoda, these structural motifs are maximal in both length and support.

Here, we demonstrate how the Gemoda algorithm can be used for structural motif discovery by 'discovering' the structural homology between the human galactose-1-phosphate uridylyltransferase (PDB id 1HXQ) (Wedekind *et al*., 1996) and fragile histidine triad proteins (PDB id 3FIT) (Lima *et al*., 1997), originally reported elsewhere (Holm and Sander, 1997). Using Gemoda, we looked for motifs of at least 30 residues, occurring in at least 3 chains, that had a pairwise RMSD of 1.5 Å or less (based on superposition of the alpha-carbon backbone) over each window of 30 residues.

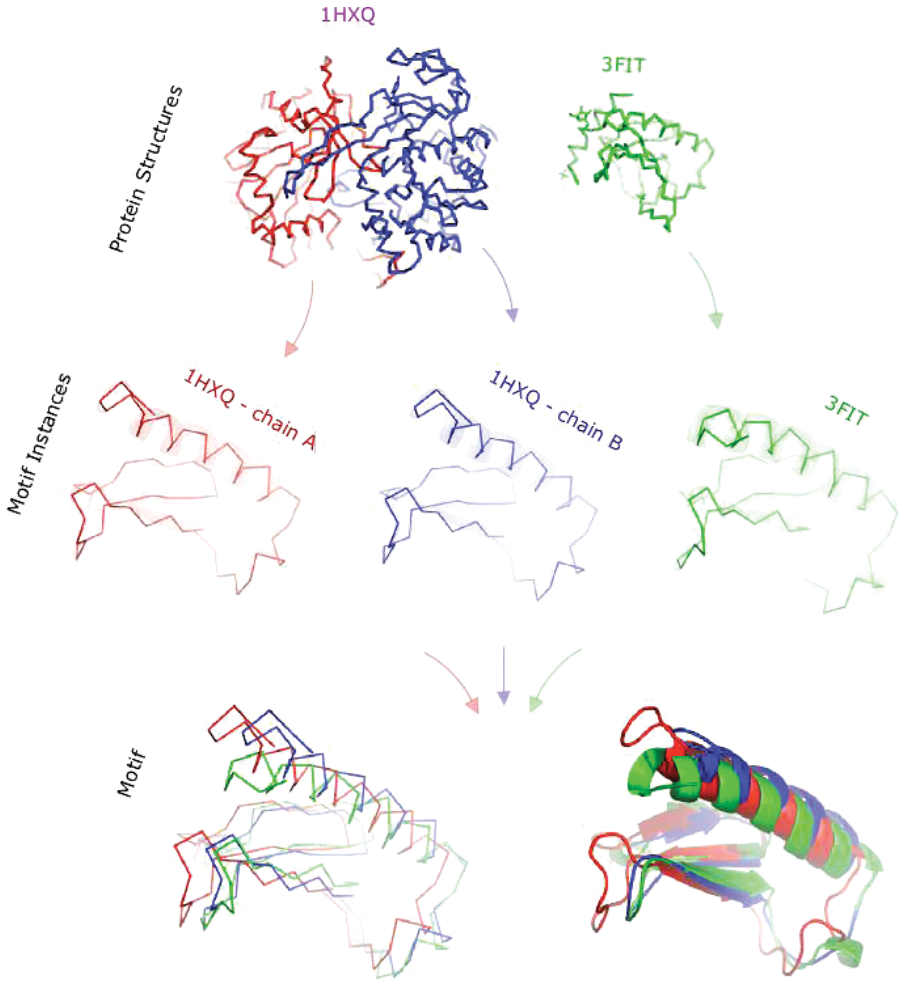

Fig. 5. A motif showing structural conservation between the human galactose-1-phosphate uridylyltransferase and fragile histidine triad proteins originally reported by Holm and Sander (1997). The motif, as shown here, was 'discovered' using the Gemoda algorithm along with three other, smaller, structural motifs that are highly conserved between the two proteins. Notably, the proteins show little sequence similarity over the region displayed in the structural motif above. Graphics created using PyMol (DeLano Scientific, San Carlos, CA, USA).

This search returns 4 motifs, the longest of which is 66 residues (Fig. 5). This motif has one embedding in the 3FIT protein and two, in different chains, in the 1HXQ protein. As shown in the figure, the motif is an alpha helix followed by a beta sheet.

DISCUSSION

Gemoda makes four contributions. First, the algorithm is generic in that it is equally applicable to any variety of sequential data. Second, Gemoda allows arbitrary similarity metrics. In the examples shown here, we chose relatively simple metrics (scoring matrices and RMSD-base metrics); however, similarity metrics can be easily changed or added. For example, in the case of amino acid sequences, one can easily define hybrid metrics incorporating primary, secondary, and tertiary structure features. In the case of nucleotide sequences, the metric may be changed to incorporate methylation information. The third contribution is that Gemoda returns motifs that are not tied to any particular motif representation. In the case of amino acid sequence motifs, it is easy to model Gemoda's motifs using regular expressions, hidden Markov models or position-specific scoring matrices. Finally, when used with the clique-finding clustering algorithm, Gemoda returns an exhaustive set of maximal motifs. To the best of our knowledge, Gemoda is the only motif discovery algorithm incorporating the above features.

As mentioned in the introduction, Gemoda integrates the best characteristics from a number of previously published motif and association discovery algorithms. For specific problems, Gemoda's performance can be improved further, though at the expense of generality. For example, a window sampling approach such as that used by Blast (Altschul *et al*., 1997) would be useful in applications where speed is more important than completeness of results. For protein structure comparisons Gemoda could also be altered to use contact maps like those used by Dali (Holm and Sander, 1993). The convolution stage could also be made faster by using heuristical, non-exhaustive convolution methods. Also, the clustering phase could be expedited by using approximate clique finding methods.

Futhermore, the Gemoda algorithm could be modified to find gapped motifs. As currently formulated, Gemoda can find motifs with short, fixed length gaps; however if a gap causes a motif to

fail to meet the similarity threshold during convolution, then it is not extended. It may be possible to alter the convolution step to allow for large or variable-length gapped motifs. Another option is to look for maximal motifs whose offsets are highly correlated. Our studies indicate that such *post hoc* analysis of Gemoda's output can usually find well-conserved gapped motifs, including those with variable gap lengths, as was the case for the (ppGpp)ase example.

Gemoda's generic nature makes it readily applicable for many problems. In the protein sequence application, Gemoda's exhaustive search using a scoring matrix as a similarity metric identified multiple motifs. It provided an accurate representation of these domains in as much as an 8-fold excess of spurious sequences. In the DNA motif discovery application, Gemoda identified an otherwise unintentional result in a synthetic dataset and satisfactorily described a motif embedded in a genomic dataset. In the protein structure application, Gemoda demonstrated that it can compare multiple arbitrary-dimensional structures simultaneously and return results previously shown in the literature. Gemoda can also be directly applied to other diverse types of sequential datasets, or it can be extended to address problems not yet considered.

Conflict of Interest: none declared.

REFERENCES

Alexandrov,N.N. (1996) SARFing the PDB. *Protein Eng.*, **9**, 727–732.

Alexandrov,N.N. and Fischer,D (1996) Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins*, **25**, 354–365.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Aravind,L and Koonin,E.V. (1998) The HD domain defines a new superfamily of metal-dependent phosphohydrolases. *Trends Biochem Sci.*, **23**, 469–472.

Arun,K. S. *et al.* (1987) Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, **9**, 698–700.

Bailey,T.L. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **2**, 28–36.

Bairoch,A (2000) The ENZYME database in 2000. *Nucleic Acids Res.*, **28**, 304–305.

Bairoch,A. and Apweiler,R. (2000) The Swiss-Prot protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Bateman,A *et al.* (2004) The Pfam protein families database. *Nucleic Acids Res.*, **32** (Database issue), D138–D141.

Buhler,J. and Tompa,M. (2001) Finding motifs using random projections. In *Proceedings of the Fifth Annual International Conference on Computational Biology*. Quebec, Canada, ACM Press, New York, pp. 69–76.

Crooks,G.E. *et al.* (2004) WebLogo: A sequence logo generator. *Genome Res.*, **14**, 1188–1190.

Dietmann,S and Holm,L. (2001) Identification of homology in protein structure classification. *Nat. Struct. Biol.*, **8**, 953–957.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Eidhammer,I. *et al.* (2000) Structure comparison and structure patterns. *J. Comput. Biol.*, **7**, 685–716.

Eskin,E. and Pevzner,P.A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, **18** ((Suppl. 1)), 354–363.

Garey,M. and Johnson,D. (1979) *Computers and Intractability: A Guide to the Theory of NP—Completeness*. W.H. Freeman and Company, NY.

Henikoff,S. and Henikoff,J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Henikoff,S. *et al.* (1995) Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene*, **163**, GC17–GC26.

Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.

Hofmann,K. *et al.* (1999) The PROSITE database, its status in 1999. *Nucleic Acids Res.*, **27**, 215–219.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.

Holm,L and Sander,C (1997) Enzyme HIT. *Trends Biochem Sci.*, **22**, 116–117.

Holm,L. *et al.* (1992) A database of protein structure families with common folding motifs. *Protein Sci.*, **1**, 1691–1698.

Horn,B.K.P. (1987) Closed-form solution of absolute orientation using unit quaternions. *J. Optical Soc. America A*, **4**, 629–642.

Hunter,C.G. and Subramaniam,S. (2003) Protein fragment clustering and canonical local shapes. *Proteins*, **50**, 580–588.

Jonassen,I *et al.* (1995) Finding flexible patterns in unaligned protein sequences. *Protein Sci*, **4**, 1587–1595.

Jonassen,I. *et al.* (2002) Structure motif discovery and mining the PDB. *Bioinformatics*, **18**, 362–367.

Keich,U. and Pevzner,P.A. (2002) Finding motifs in the twilight zone. *Bioinformatics*, **18**, 1374–1381.

Kolodny,R. *et al.* (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J. Mol. Biol.*, **346**, 1173–88.

Lawrence,C.E. *et al.* (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **262**, 208–214.

Lima,C.C. *et al.* (1997) MAD analysis of FHIT, a putative human tumor suppressor from the HIT protein family. *Structure*, **5**, 763–774.

Madej,T *et al.* (1995) Threading a database of protein cores. *Proteins*, **23**, 356–369.

Mancheron,A and Rusu,I. (2003) Pattern discovery allowing wild-cards, substitution matrices, and multiple score functions. In *Algorithms in Bioinformatics, Proceedings of the Lecture Notes in Bioinformatics. Algorithms in Bioinformatics: Third International Workshop, WABI 2003*, Budapest, Hungary, Springer-Verlag, Berlin, pp. 124–138.

Marchler-Bauer,A. *et al.* (2003) CDD: a curated Entrez database of conserved domain alignments. *Nucleic Acids Res.*, **31**, 383–387.

Murthy,V.L. and Rose,G.D. (2003) RNABase: an annotated database of RNA structures. *Nucleic Acids Res.*, **31**, 502–504.

Orengo,C.A. and Taylor,W.R. (1996) SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol*, **266**, 617–635.

Ortiz,A.R. *et al.* (2002) MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.*, **11**, 2606–2621.

Pevzner,P.A. and Sze,S. (2000) Combinatorial Approaches to finding subtle signals in DNA sequences. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, pp. 269–278.

Pevzner,P. and Sze,S.H. (2001) private communication.

Price,A *et al.* (2003) Finding subtle motifs by branching from sample strings. *Bioinformatics*, **19** (Suppl. 2), II149–II155.

Rigoutsos,I. *et al.* (1999) Dictionary building via unsupervised hierarchical motif discovery in the sequence space of natural proteins. *Proteins*, **37**, 264–77.

Rigoutsos,I. and Floratos,A. (1998) Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, **14**, 55–67.

Salgado,H. *et al.* (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res.*, **32** (Database issue), D303–D306.

Styczynski,M. *et al.* (2004) An extension and novel solution to the motif challenge problem. *Genome Informatics*, **15**, 63–71.

Tomita,E. *et al.* (1989) An Optimal Algorithm for finding all the cliques. *SIG Algorithms*, **12**, 91–98.

Tompa,M. *et al.* (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.*, **23**, 137–144.

Wedekind,J.E. *et al.* (1996) The structure of nucleotidylated histidine-166 of galactose-1-phosphate uridylyltransferase provides insight into phosphoryl group transfer. *Biochemistry*, **35**, 11560–11569.

Zaki,M.J. and Ogihara,M. (1998) Theoretical foundations of association rules. In *Proceedings of 3rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98)*. Seattle, Washington, pp. 7:1–7:8.

Zaki,M.J. (2000) Scalable algorithms for association mining. *Knowledge Data Eng.*, **12**, 372–390.