

Chapter 7

Improved Algorithms for Protein Motif Recognition

Bonnie Berger *#

David B. Wilson *+

Abstract

The identification of protein sequences that fold into certain known three-dimensional (3D) structures, or motifs, is evaluated through a probabilistic analysis of their one-dimensional (1D) sequences. We present correlation methods that run in linear time and incorporate pairwise dependencies between amino acid residues at multiple distances to assess the conditional probability that a given residue is part of a given 3D structure. One of these methods is generalized to multiple motifs, where a dynamic programming approach leads to an efficient algorithm that runs in linear time for practical problems. By this approach, we were able to distinguish (2-stranded) coiled-coil from non-coiled-coil domains and globins from nonglobins.

1 Introduction

The so-called “grand challenge” problem associated with protein folding is to determine how a protein will fold in 3-dimensions when given only its amino acid sequence. In addressing this problem, there are three relevant structural classes of protein information. The 1D structure is the amino acid sequence of the protein. Secondary structure is common recurring local folding patterns, such as α -helices, β -sheets, and coiled coils. Tertiary structure is the complete global fold of a protein, including the positions of the backbone and side-chain atoms, with their corresponding bond lengths, bond angles, and torsional angles. The fold of a protein provides the key to understanding possible biological function.

An important first step in tackling the protein folding problem, which is already useful in its own right, is a solution to what we call the *motif recogni-*

tion problem, also called *profile analysis* in the biological literature [33, 17, 27, 26]. The motif recognition problem is: given a known 3D structure, or motif, determine whether this fold occurs in a given amino acid sequence, and if so, in what positions. In most cases, the motif is a region of secondary structure.

Many soluble protein sequences can, with some difficulty, be synthesized and tested in the laboratory to see if they fold into a given motif; however, such a process is expensive and can take up to one to two months per sequence. Therefore, a computer program that recognizes motifs solely from sequence data can expedite recognition and guide understanding of the protein folding problem [9].

The motif recognition problem is currently an active area of computational research. The goal is to develop a sieve: a way of separating sequences that fold into a particular motif from those that do not. We introduce a general framework for motif recognition, based solely on the “right” correlation probabilities, which seems to do as well or better than more complicated algorithms. Our algorithms have the following advantages:

- They are a natural generalization of the methods that assume complete independence of amino acid residues [29, 17, 27, 26, 15], already in use by biologists.
- Unlike *hidden Markov models* [18, 23, 4, 12], they compute a likelihood score for each residue position in a given sequence, rather than a single final probability for the entire sequence. This extra data is valued by biologists for their understanding of structure [21].
- Unlike hidden Markov models, they naturally handle pairwise correlations between amino acid residues at multiple distances simultaneously. This has been found to be essential for the successful recognition of certain motifs, such as coiled coils [6].

*Department of Mathematics and Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

#Supported in part by a grant from the MIT-State Street Bank Science Partnership Fund.

+Supported in part by an ONR-NDSEG fellowship.

- They can be proved to compute the right answer when the sequences in the database are assumed to satisfy a basic set of minimal assumptions.
- A dynamic programming algorithm is presented that generalizes these methods to multiple motifs and runs in linear time for practical problems.

The best reasons to use our algorithms, however, are that they are relatively simple to code, run very efficiently (linear-time for most variants), and have already been implemented to successfully recognize coiled coils [6] and globins.

Related Work

We now survey previous work in the biological and computational communities. Two main approaches, or *template methods*, have been pursued in the biological literature. The feature that distinguishes these two approaches is whether they make use of tertiary structure information.

1D methods: These use a statistical analysis of the sequence, sometimes using known secondary structure, to correlate sequence positions. These techniques range from sequence alignments [10, 20, 2] to tailored approaches, based on properties of the amino acid residues that can be deduced from the sequence and secondary structure (*e.g.*, hydrophobicity, size, solvent accessibility) [29, 25, 5, 7, 26]. These methods have been moderately successful at motif recognition but have been unable to completely distinguish positive from negative examples of a fold.

Researchers have started to investigate pairwise correlations between amino acid residues. Pairwise correlations were found between particular pairs of residues in the zinc finger binding domain [13] and β -sheets [32, 24], and between pairs of tRNA nucleotides [22, 1]. Recently, researchers have focused on hidden Markov models [18, 23, 4, 12], which can take into account consecutive pairs of residues. Using these methods, Haussler *et al.* report they achieve good recognition of globins. Our approaches have some similarities and some important differences with hidden Markov models (see discussion below).

3D methods: There has also been a considerable amount of recent work in the direction of using geometric and physical models of protein structure to predict folding [30, 19, 16, 31, 8]. This work incorporates pairwise correlations between residues that are physically close. These more complicated methods require the full tertiary structure of many different folds to be known, and discovering the latter is a slow and costly biological endeavor. In fact, only about two to three hundred distinct folds are currently known [21].

Our Results

We present a linear-time algorithm for general motif recognition and prove that it works, given plausible probabilistic assumptions on the sample data. Our 1D method is strong enough to capture all pairwise correlations between amino acid residues. This includes non-fixed-distance-based pairs, as well as more complicated functions such as minimums or averages of multiple pairs of residues, as well as pairs of residues at fixed distance d apart. Implementing multiple pairs simultaneously is what allowed us to obtain complete separation between positive and negative examples of coiled-coil domains [6]; in particular, the *minimum* over multiple pair scores seems to work very well since it deters against “false positives”.

The method can be extended to recognize multiple motifs in the same sequence, which are ordered and distinct, by a new dynamic programming algorithm. The algorithm aligns m regions of secondary structure within a sequence of length n in $O(mn)$ time, independent of the imposed maximum gap size between motif regions. This is a linear-time speedup over the running time of standard methods.

Our more sophisticated algorithm has the desirable property that it runs in linear time on problems with a constant number of motifs. Typical applications in practice have a small (*i.e.*, constant) number of motifs. Prior implementations used in practice, such as the well-known Lupas *et al.* [25] program for recognizing coiled coils and the Bashford *et al.* [5] program for recognizing globins, required over a day to run on large databanks such as *Genpept*. (*Genpept* is the database of nearly all available protein sequences, translated from *GenBank*.) Using the improved algorithms described in this paper, we are able to run through the entire *Genpept* in approximately 15 minutes on a Sun SPARC 10 computer. Hence, in addition to giving better performance in terms of our output on coiled coils and globins, our algorithms run substantially faster than those used in practice.

The first algorithms we present for single and multiple motifs are different in approach from hidden Markov models. Unlike the Markov models, they naturally handle pairwise correlations between amino acid residues at multiple distances simultaneously. This has been found to be essential for recognizing certain motifs successfully, such as coiled coils [6]. The Markov methods do not seem to have been generalized successfully to non-consecutive pairs of residues. In fact, using the natural Markov approach on multiple-distance pairs simultaneously would require statistics involving correlations between three or more residues, where pairs suffice for us. There does not appear to be enough data to handle correlations involving

triples of residues. In addition, our algorithms retain more information by means of a score for each residue position, rather than an overall alignment score for the entire sequence. When analyzing a particular protein of interest, biologists find the likelihood score for each position to give valuable information as to the relative “strength” of the motif in each location [21].

Even though our correlation functions do not correspond to hidden Markov models in the natural sense, we show toward the end of the paper (Section 4) that we could in fact generalize simple hidden Markov models, and then adapt the correlation method of Section 2 to such a Markov framework. This second method gives an alternative linear-time algorithm, which is best understood as a generalization of the Markov methods to correlations between multiple-distance pairs of residues. The second algorithm could also be seen as using our correlation functions to generalize the dynamic programming approach introduced in [15] to pairwise correlations.

In practice, the main advantage of the first correlation method we present is that it returns a score for each residue; the advantage of the second is that it is a “windowless” approach, which may be preferable for the recognition of certain motifs.

Implementation

The first correlation method has been implemented to recognize both (2-stranded) coiled coils [6] and globins. In [6], we applied the method to predict coiled-coil domains in protein sequences by using pairwise-residue correlations obtained from a (2-stranded) coiled-coil database of 58,635 amino acid residues. The implementation recognized every position in the distinct coiled coils that were used to make the database. Similarly, it obtained over 99.5% recognition of the 610 different globins tested. The most popular coiled coil prediction scheme [25], when tested with the Brookhaven X-ray crystal structure database, has a “false positive” prediction rate of 7% (14 out of the ~ 215 nonhomologous proteins are predicted to have coiled coils when their X-ray crystal structures show that no coiled coil exists). These false positives represent 2/3 of the proteins their method predicts to have coiled coils. Usually these false positive predictions correspond to amphipathic α -helices that are not coiled coils. By contrast, our method has no false positives or false negatives when it is tested with the Brookhaven database. 1D methods were previously believed incapable of achieving such success. In fact, for α -helices, a 66% success rate has been regarded as very good.

The location of a new coiled coil, found by the method, has already been corroborated in the laboratory. Namely, the method identified a coiled-

coil region in the moloney murine leukemia virus, whose existence has since been verified [14]. The method has identified similar coiled-coil regions in influenza hemagglutinin and HIV. These coiled-coil regions are important because the formation of coiled coils has recently been found to be the mechanism by which the influenza virus hemagglutinin binds to the cell membrane [9].

A companion paper [6] reporting experimental data based on the correlation method has been submitted for journal publication.

2 The Correlation Method

We present a method for determining whether an unknown sequence contains a given motif in its fold, and if so, in what positions. First let us define some terms. Let \mathbf{C} be the class of positive examples; for our purposes, this is the set of all w -long sequences in Genpept (the database of nearly all available protein sequences) that fold into the given motif. Suppose we have a *window* of fixed length w . We say that a window is *aligned* to a sequence when it is positioned over a contiguous subsequence of length w .

In our applications, since there is a lower bound on the length of the motif [28, 5], a fixed size window suffices. Our algorithm’s running time will be independent of the window size. The window-based approach has the advantage that it computes a likelihood score for each residue position. However, in Section 4, we adapt the correlation method to a “windowless” approach.

In this section, we present algorithms for finding the best alignment of a single motif pattern to a given sequence, and in Section 3 show how to use dynamic programming to generalize this to multiple, ordered motif patterns in a given sequence.

The basis for the correlation methods presented in this section is computing pairwise correlations between amino acid residues some constant distance d apart in the sequence. These correlations alone can be used to recognize motifs, or alternatively, we also show how to compute more complicated functions of multiple-distance pairs. In particular, a scoring function based on the *minimum* of a number of different fixed distances seems to work well in practice, since it will deter against false positives [6].

2.1 Single Distance d Correlations

For each alignment of a w -long window to a given sequence, we wish to compute an *estimate* for the probability that the aligned subsequence is in \mathbf{C} . We then take the maximum-valued alignment over the entire sequence to estimate the probability that some subsequence is in \mathbf{C} , or alternatively, that a particular

amino acid residue is in a subsequence which is in \mathbf{C} . Finally, in Section 2.3, we use this probability to interpret a score for a given sequence (or residue).

How do we compute the probability estimate for a particular alignment? Traditionally, researchers estimated this probability by tabulating the single residue frequencies in the window under the implicit assumption that they were independent of each other. We find that using conditional higher dependencies between pairs of residues provides a more effective sieve; that is, a way of separating sequences (or residues) that are in \mathbf{C} from those that are not. First, we consider pairs of residues some constant distance d apart in 1D sequence (where distance $d = 1$ corresponds to consecutive positions).

Genpept defines the probability distribution over the universe of protein sequences. We are assuming all probabilities are with respect to the Genpept probability distribution.

Suppose we are given sequence $z = r_1 r_2 \cdots r_w$. Let $x = R_1 R_2 \cdots R_w$ be a sequence randomly drawn from Genpept, G_i be the event that $R_i = r_i$, and H_i be the event $G_i \wedge G_{i+d}$. Given any sequence y , also define C_y to be the event that y is in \mathbf{C} . Note that, for all z , $\Pr[C_z]$ is one or zero, depending on whether or not z is in \mathbf{C} . We will estimate the conditional probability $\Pr[C_x | G_1 \wedge G_2 \wedge \cdots \wedge G_w]$, which is $\Pr[C_x]$.

We assume dependencies between only pairs of residues distance d apart. More formally, for all k , we assume that

$$(2.1) \quad \Pr[G_k | C_x \wedge G_{k+1} \wedge \cdots \wedge G_w] = \Pr[G_k | C_x \wedge G_{k+d}]$$

and

$$(2.2) \quad \Pr[G_k | G_{k+1} \wedge \cdots \wedge G_w] = \Pr[G_k | G_{k+d}],$$

where the event G_{k+d} is true if $k+d > w$. That is, for sequences randomly drawn from \mathbf{C} or Genpept, we assume that the probability of G_k is dependent only on G_{k+d} among $\{G_{k+1}, \dots, G_w\}$. Note that we are not assuming G_k is independent of the other residues, just that whatever dependency there is is captured in G_{k+d} . Under these assumptions, the following theorem gives the probability that sequence x is in \mathbf{C} , conditioned on the residues that appear in x .

THEOREM 2.1. (PAIR FREQUENCY METHOD)
Assuming Equations 2.1 and 2.2,

$$\begin{aligned} \Pr[C_x] &= \Pr[C_x | G_1 \wedge \cdots \wedge G_w] \\ &= \Pr[C_x] \cdot \frac{\prod_{i=1}^{w-d} \Pr[H_i | C_x]}{\prod_{i=1+d}^{w-d} \Pr[G_i | C_x]} \cdot \frac{\prod_{i=1+d}^{w-d} \Pr[G_i]}{\prod_{i=1}^{w-d} \Pr[H_i]}. \end{aligned}$$

PROOF: The first term in the right-hand-side is the probability that x is in \mathbf{C} ; the second

term is the product of the probabilities of pairs of residues occurring in \mathbf{C} divided by the product of the probabilities of the single residues repeated in the pairs occurring in \mathbf{C} ; and the third term is the reciprocal of the analogous pair to single probability ratio in Genpept.

By Bayes' Law,

$$\Pr[C_x | G_1 \wedge \cdots \wedge G_w] = \frac{\Pr[C_x \wedge G_1 \wedge \cdots \wedge G_w]}{\Pr[G_1 \wedge \cdots \wedge G_w]}.$$

We first focus on expanding the numerator. Thus, by repeated application of Bayes' Law,

$$\begin{aligned} &\Pr[C_x \wedge G_1 \wedge \cdots \wedge G_w] \\ &= \Pr[G_1 | C_x \wedge G_2 \wedge \cdots \wedge G_w] \cdot \Pr[C_x \wedge G_2 \wedge \cdots \wedge G_w] \\ &= \Pr[G_1 | C_x \wedge G_{1+d}] \cdot \Pr[C_x \wedge G_2 \wedge \cdots \wedge G_w] \\ &= \vdots \quad (\text{by repeated applications of Bayes' Law} \\ &\quad \text{and then Equation 2.1}) \\ &= \prod_{i=1}^w \Pr[G_i | C_x \wedge G_{i+d}] \cdot \Pr[C_x] \\ &= \prod_{i=1}^{w-d} \frac{\Pr[G_i \wedge G_{i+d} \wedge C_x]}{\Pr[G_{i+d} \wedge C_x]} \cdot \prod_{i=w-d+1}^w \Pr[G_i | C_x] \cdot \Pr[C_x] \\ &= \frac{\prod_{i=1}^{w-d} \Pr[H_i | C_x]}{\prod_{i=1+d}^{w-d} \Pr[G_i | C_x]} \cdot \Pr[C_x] \quad (\text{by def. of } H_i). \end{aligned}$$

By a similar analysis, using Equation 2.2 instead of Equation 2.1, it can easily be shown that the denominator is

$$\Pr[G_1 \wedge \cdots \wedge G_w] = \frac{\prod_{i=1}^{w-d} \Pr[H_i]}{\prod_{i=1+d}^{w-d} \Pr[G_i]}.$$

Combining the equations for the numerator and denominator, we obtain the desired theorem. \square

Using the notation of Theorem 2.1, consider the special case where the G_i 's are completely independent.

COROLLARY 2.1. (SINGLE FREQUENCY METHOD)
Assuming the G_i 's are completely independent,

$$\begin{aligned} \Pr[C_x] &= \Pr[C_x | G_1 \wedge \cdots \wedge G_w] \\ &= \Pr[C_x] \cdot \prod_{i=1}^w \Pr[G_i | C_x] \cdot \frac{1}{\prod_{i=1}^w \Pr[G_i]}. \end{aligned}$$

We note that this gives a formal framework for interpreting the single frequency methods of [29, 25, 7, 26], which compute a score function of $\prod_{i=1}^w \Pr[G_i | C_x] / \prod_{i=1}^w \Pr[G_i]$. By the corollary, this score, times the constant factor $\Pr[C_x]$, is just $\Pr[C_x | G_1 \wedge \cdots \wedge G_w]$.

Estimating the individual probabilities. We use the above theorem to estimate $\Pr[C_z]$ by estimating the individual probability terms in the theorem. We use sample data to approximate the probability terms. (See Appendix A.)

2.2 Multiple-distance Correlations

Variations on the correlation method that are based on dependencies between residue pairs that are not a uniform fixed distance d apart can be constructed in a similar manner. Theorem 2.1 generalizes to correlations between pairs i and $\sigma(i)$, where σ is an arbitrary function of i . The G_i 's are the same, but the H_i 's now correspond to events $G_i \wedge G_{\sigma(i)}$, where $\sigma(i)$ is the new position paired with i . Consequently, the assumptions for Theorem 2.1 are now somewhat different: the right-hand-side of Equation 1 is now $\Pr[G_k | C_x \wedge G_{\sigma(k)}]$, and the right-hand-side of Equation 2 is now $\Pr[G_k | G_{\sigma(k)}]$. The proof of the theorem is similar. As before, if $\sigma(k) > w$, then $G_{\sigma(k)}$ is taken to be true. Consequently, the numerator is

$$\frac{\prod_{i:1 \leq i < \sigma(i) \leq w} \Pr[H_i | C_x]}{\prod_{i:\sigma(i) \leq w} \Pr[G_{\sigma(i)} | C_x]} \cdot \prod_{i:\sigma(i) > w} \Pr[G_i | C_x] \cdot \Pr[C_x].$$

The denominator is solved similarly.

Another of the many possible variations, which seems to work well in practice since it deters against false positives, is a combination of methods based on pairs at distances d and d' .

$$\Pr[C_x | \wedge_{i=1}^w G_i] = \min\{\Pr[C_x | \wedge_{i=1}^w G_i]_d, \Pr[C_x | \wedge_{i=1}^w G_i]_{d'}\},$$

where the subscripts on the probabilities correspond to dependent pairs at distance d and d' , respectively. The probabilities are then computed as before. In other words, we choose the lower of the scores achieved by the two methods. The final score can be viewed as a statistical point in 2D space. Dependencies between pairs of neighbors at L different distances likewise can be represented as a point in L -dimensional space. Experimentation will be needed to determine whether minimum, average, or some other function will produce the best sieve for a particular motif.

The methods in this paper can be further extended to handle k -wise dependencies for 1D template methods for any k . *A priori*, it was not clear that pairwise correlations or correlations for $k > 2$ should help. However, our results indicate that pairwise correlations are sufficient to separate (2-stranded) coiled-coil from non-coiled-coil domains [6] and globins from nonglobins. Strictly greater than pairwise correlations may indeed improve 1D methods, but there does not seem to be enough data to support this approach. The question remains whether pairwise or greater correlations will increase the power of 3D template methods.

2.3 The Algorithm

Given a sequence of amino acid residues, let z be the subsequence restricted to a w -long window. We use one of the correlation methods above to compute an estimate for the probability that the subsequence z is in C . Let P_z be the resulting probability estimate.

We begin by interpreting the probability estimate as a score. Set $S_z = \log P_z - \log \Pr[C_x]$ to be the score for z (*i.e.*, the score S_z is simply the logarithm of the product of the last two terms on the right-hand-side of the equation in Theorem 2.1).

To determine all the regions of a sequence that fold into a given motif C , we need only score all the residue positions. In this section, we show how to do this for periodic motifs (see Section 4). Many motifs, such as α -helices, β -sheets, and coiled coils, are periodic. The next section handles aperiodic motifs.

The algorithm glides the w -long window from left to right along the sequence, maintaining a running sum and keeping track of the maximum score for each residue. We thus obtain the following theorem.

THEOREM 2.2. *The running time to compute a score for a sequence of length n is $O(pn)$, where p is the period of the motif. The running time does not grow with the window length w . For periodic motifs, the running time is $O(n)$ for all practical purposes.*

PROOF: Because we keep a running sum, each position is visited only when the window first glides over it and when the window exits it. Computing the maximum score for each residue, independent of window size, also takes constant time per position by using a method similar to the block method described in the next section. Therefore, the total time to glide the window over the sequence and do the necessary computations is $O(n)$. This is done p times. \square

Now, the question is, what is a high score? One can give an estimate for this using the methods in Sections 2.1 and 2.2, based upon an estimate for $\Pr[C_x]$. Another way to determine a high score is to run the algorithm on a bunch of positive and negative samples [6] and then pick a good separating score. Since $\log \Pr[C_x]$ is added to all the scores, it is not important for comparison purposes. Since an estimate of $\Pr[C_x]$ is generally not known, the latter approach has proven to be most reliable in practice.

3 Alignment of Multiple, Ordered Motifs

If we want to align multiple motifs that have a specified ordered arrangement, we may turn to dynamic programming. The following dynamic programming algorithm was inspired by the globins [5], which are oxygen-carrying proteins such as myoglobin and hemoglobin. These have a number of different ordered motifs, and each is always within a certain fixed

distance of the next. If a negative example has one of the motifs, followed by the succeeding motif many positions later than the maximum gap between these would allow, it is assumed that it is not a globin. In general, it is very often the case with proteins that we know certain regions of secondary structure should occur within a certain range of each other.

Given a sequence of length n , we want to find a maximum scoring alignment of fixed-size contiguous windows B_1, \dots, B_m ($\sum_{i=1}^m |B_i| \leq n$) to the sequence from left to right in increasing order of i , such that the windows are non-overlapping, windows i and $i+1$ are placed within at most g_i positions of each other, and every window position aligns to some sequence position. The score of an alignment is the sum of its individual window scores, computed by the correlation methods in Section 2. Our solution to this problem uses dynamic programming to achieve an $O(mn)$ algorithm, independent of the maximum gap sizes g_i . Since m is typically constant, as with the globins ($m = 7$), this gives us a linear-time algorithm.

If we used the naive implementation for this dynamic programming algorithm, it would run fairly slowly ($O(mn^2)$ time), since the algorithm would have to “look back” $O(n)$ for each entry in the $m \times n$ matrix. However, we can get around this problem by partitioning the columns of the matrix into equal-size blocks. This partition allows us to perform single left and right sweeps across the rows, using this information to efficiently compute maximum scores, even with the gap restrictions between the windows.

The dynamic programming matrix has m rows and n columns. Define $M(i, j)$ to be the maximum score for the first i windows when the i th window ends in some position $k \in [j - g_i, j]$. Let $M(0, j) = 0$ for all j . For all matrix positions (i, j) , let

$$N(i, j) = \text{score}(i, j) + M(i - 1, j - |B_i|),$$

where $\text{score}(i, j)$ is the score of the i th window ending at position j .

For all matrix positions (i, j) , compute $M(i, j)$ from $N(i, j)$ as follows. For a given i , partition the n columns of N into $\lceil n/g_i \rceil$ contiguous blocks of size g_i (except the last one is $\leq g_i$ in size.) Then for each position j , let $N_L(i, j)$ be the maximum $N(i, k)$, where k ranges from the beginning of j 's partition block to position j , and let $N_R(i, j)$ be the maximum $N(i, k)$, where k ranges from the end of j 's partition block to position j . $N_L(i, j)$ ($N_R(i, j)$) can be computed by a single left-to-right (right-to-left) scan over the i th row of N , starting a new maximum each time a barrier of a partition block is crossed. Compute

$$M(i, j) = \max\{N_L(i, j), N_R(i, j - g_i)\}.$$

Notice that we save computing $\max_{j-g_i \leq k \leq j} N(i, k)$ for every matrix entry. The maximum score is $M(m, n)$. The actual alignment is found using standard methods [11].

THEOREM 3.1. *Given the score matrix, the dynamic programming algorithm aligns m non-overlapping windows to a sequence of length n in $O(mn)$ time, independent of the maximum gap sizes g_i .*

PROOF: Given the score matrix, $N(i, j)$ can be computed in constant time once its M values are computed. N_L and N_R take $O(n)$ time for each row; thus, it takes $O(mn)$ time overall to compute $M(m, n)$, the maximum score. Recovering the actual alignment takes linear time. \square

If the different motifs are periodic, the score matrix can be computed with a running sum through a left to right scan of the sequence in $O(n \sum_{i=1}^m p_i)$ time, where p_i is the period of motif B_i . Note that for all practical purposes this running time is $O(nm)$. If the motifs are aperiodic, then the running time is $O(n \sum_{i=1}^m |B_i|)$. We pose as an open problem whether the running time can be improved, for example, by using convolution techniques, developed in pattern matching, to compute the scores for the windows.

4 A Markov Framework for the Correlation Method

Some protein motifs are periodic, for example, α -helices, β -sheets, and especially coiled coils. Consider the coiled-coil motif. Coiled coils consist of α -helices wrapped around each other in a superhelical twist. Since these α -helices have about 3.5 residues per turn, coiled coils have a characteristic heptad repeat (**abcdefg**) $_n$. Coiled coils are biologically important as they are found in fibrous proteins such as myosin, several DNA binding domains, some tRNA synthetases, tumor suppressor gene products, and membrane fusion proteins.

Given a random protein, the method described below outputs the “most probable” assignment of the amino acids to positions in the motif. For coiled coils, the motif positions are the symbols **a** through **g**, together with a special symbol **z** to represent residues not in a coiled coil. For example, the experimentally determined assignment for human keratin is

amino acids: ... **EIATYRRLLEGEDAHLS** ...
 motif positions: ... **gabcdefgabcdzzzzz** ...

To find the most likely assignment, we could compute for each string S of motif positions the probability p_S that this string characterizes the residues of a random protein. Next compute the probability q_S that a random protein characterized by S will be the

given protein. The output is the string S which maximizes $p_S q_S$. Note that the given sequence can be represented in about $\lg[1/p_S q_S]$ bits by writing down the sequence S of motif positions and then the amino acid sequence. The output minimizes the number of bits needed to represent the protein in this way.

The algorithm described below uses a Markov chain model to estimate p_S and statistical information to estimate q_S . The optimal string of motif positions is computed with dynamic programming.

In [15], an efficient pattern-matching algorithm is presented for identifying coiled coils and other periodic protein motifs. The method described here may be viewed as a generalization of their algorithm to handle pairwise residue correlations. The pattern-matching gap penalties would be determined by the Markov chain state transition probabilities. There are also differences in how the dynamic programming is carried out.

4.1 The Method

We demonstrate our method as applied to coiled coils. Each residue in a sequence of amino acids will be in one of eight states depending on whether or not it is in a coiled coil, and if so in which position. For example, if an amino acid residue is in the **c** position, chances are that the next residue will be in position **d**; if a residue is not in a coiled coil (referred to as position **z**), chances are that the next one will not be in a coiled coil as well (see Figure 1). For convenience, let ν denote the permutation defined by these likely transitions. But

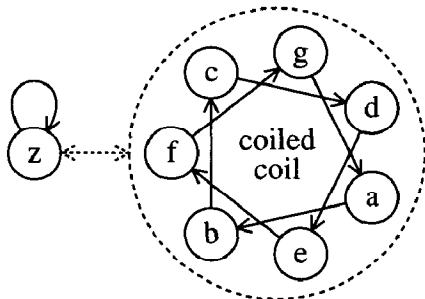


Figure 1: Coiled-coil states. State transitions follow the solid arrows with probability close to 1 (*i.e.*, the solid lines define the permutation ν). Other state transitions sometimes occur, but only the transitions from non-coiled-coil to coiled-coil states and back are depicted here.

since proteins containing coiled coils often contain non-coiled-coil regions, transitions from coiled-coil states to **z** and back sometimes occur. Irregularities known as “skips” and “stutters” sometimes occur in a coiled coil. They are found in places such as

the hinge region of myosin. In these locations, a residue in coiled-coil position l is followed by a residue in a coiled-coil position other than $\nu(l)$. “Phase shifts”, the state transitions corresponding to skips and stutters, are not shown in Figure 1.

More generally, the motif has period p , and the positions in the motif are labeled m_0, \dots, m_{p-1} . For coiled coils, $p = 7$ and m_0, \dots, m_{p-1} are the heptad-repeat positions **a, b, c, d, e, f, g**. Given a protein, the goal is to assign to each residue a label in $\{m_0, \dots, m_{p-1}, \mathbf{z}\}$, where **z** represents a residue which is not in the motif. In particular, for a given amino acid sequence, we want the most plausible such assignment.

Suppose the given protein consists of residues r_1, \dots, r_n . Let R_1, \dots, R_n denote a “random protein” of length n , and let L_1, \dots, L_n denote its labeling. We are seeking the labeling l_1, \dots, l_n of the given residues which maximizes the probability

$$\Pr[R_1 = r_1 \wedge \dots \wedge R_n = r_n \wedge L_1 = l_1 \wedge \dots \wedge L_n = l_n].$$

For convenience, let \mathcal{R}_k denote the event $R_k = r_k$ and \mathcal{L}_k denote the event $L_k = l_k$. ($\mathcal{R}_k = G_k$ from Section 2.)

We have by Bayes Law,

$$(4.3) \quad \Pr[\mathcal{R}_1 \wedge \dots \wedge \mathcal{R}_n \wedge \mathcal{L}_1 \wedge \dots \wedge \mathcal{L}_n] \\ = \Pr[\mathcal{L}_1 \wedge \dots \wedge \mathcal{L}_n] \Pr[\mathcal{R}_1 \wedge \dots \wedge \mathcal{R}_n | \mathcal{L}_1 \wedge \dots \wedge \mathcal{L}_n].$$

We need to estimate the first and second terms on the right-hand side of the above equation. To estimate the first term of Equation 4.3, we need some knowledge about the domain of interest. For coiled coils, we can model the label probabilities with a Markov chain approximation:

$$(4.4) \quad \Pr[\mathcal{L}_k | \mathcal{L}_1 \wedge \dots \wedge \mathcal{L}_{k-1}] \approx \Pr[\mathcal{L}_k | \mathcal{L}_{k-1}].$$

Returning to coiled coils, for lack of good biological data, we assume that the transitions from **z** to the coiled-coil states are equally likely, as are the transitions from the coiled-coil states to **z**. About 2% of residues in Genpept are in coiled coils, and the average length of the coiled coil is about 40 residues. Thus the probability that the first state is a coiled-coil state is $1/50$. Furthermore, given that a residue is in a coiled-coil state, the probability that the next residue is labeled **z** is $1/40$, and the probability of going from **z** to a coiled-coil state is $(1/40)(1/49)$. The skip and stutter transition probabilities remain to be estimated. Our experience is that skips and stutters are about as common as transitions from coiled-coil to non-coiled-coil regions, and for lack of better data, we assume that all possible phase shift transitions are

equally likely. So the phase shift transition probabilities are $(1/40)(1/6)$. To summarize:

$$\Pr[L_1 = l_1] = \begin{cases} \frac{49}{50} & l_1 = \mathbf{z} \\ \frac{1}{50} \frac{1}{7} & \text{otherwise} \end{cases}$$

$$\begin{aligned} & \text{and } \Pr[L_k = l_k | L_{k-1} = l_{k-1}] \\ & = \begin{cases} \frac{1}{40} \frac{1}{49} \frac{1}{7} & l_{k-1} = \mathbf{z} \text{ and } l_k \neq \mathbf{z} \\ 1 - \frac{1}{40} \frac{1}{49} & l_{k-1} = l_k = \mathbf{z} \\ \frac{1}{40} & l_{k-1} \neq \mathbf{z} \text{ and } l_k = \mathbf{z} \\ \frac{1}{40} \frac{1}{6} & l_{k-1} \neq \mathbf{z} \text{ and } l_k \neq \mathbf{z}, \nu(l_{k-1}) \\ 1 - \frac{2}{40} & l_{k-1} \neq \mathbf{z} \text{ and } l_k = \nu(l_{k-1}) \end{cases} \end{aligned}$$

Now suppose that the labeling of the residues is given, and we need to estimate the second term of Equation 4.3. We can do this by using the statistical information in a database of coiled coils and the Genpept database. First break up the sequence into blocks. Residues i and $i+1$ are in the same block if and only if $l_{i+1} = \nu(l_i)$. The database does not contain much information about the correlations of residues on either side of a block boundary, so we will assume that the residues in one block are independent of the residues (and labels) in other blocks:

$$(4.5) \Pr[\mathcal{R}_k | \mathcal{R}_1 \wedge \cdots \wedge \mathcal{R}_{k-1} \wedge \mathcal{L}_1 \wedge \cdots \wedge \mathcal{L}_n] \\ \approx \Pr[\mathcal{R}_k | \mathcal{R}_j \wedge \cdots \wedge \mathcal{R}_{k-1} \wedge \mathcal{L}_j \wedge \cdots \wedge \mathcal{L}_k]$$

where j is the beginning of the block containing k . Next we need an estimate for the right-hand-side of this equation. For coiled coils, the residues physically close to r_k in 3-space are those at distances 1, 3, and 4, so of r_j, \dots, r_{k-1} , the residues with the greatest influence on R_k are r_{k-1} , r_{k-3} , and r_{k-4} . (In Genpept, the residues are largely pairwise independent with the exception of positive correlations between like residues which are nearby. So for convenience, we can use distances 1, 3, and 4 for predicting R_k whether or not $l_k = \mathbf{z}$.) Let

$$P_d = \Pr[\mathcal{R}_k | \mathcal{R}_{k-d} \wedge \mathcal{L}_{k-d} \wedge \cdots \wedge \mathcal{L}_k];$$

P_d is the probability that $R_k = r_k$ if this is independent of all previous residues other than R_{k-d} . Since the database is not large enough to get good information on more than pairwise correlations, we *average* the relevant pairwise probabilities:

$$(4.6) \Pr[\mathcal{R}_k | \mathcal{R}_j \wedge \cdots \wedge \mathcal{R}_{k-1} \wedge \mathcal{L}_j \wedge \cdots \wedge \mathcal{L}_k] \\ \approx \begin{cases} P_1 & k-3 < j \leq k-1 \\ [P_1 P_3]^{1/2} & j = k-3 \\ [P_1 P_3 P_4]^{1/3} & j \leq k-4 \end{cases}$$

Assuming all these approximations are accurate, we next need to find an optimal choice for l_1, \dots, l_n . We can do this with dynamic programming in $O(n)$ time for fixed p and bounded correlation distance D . (I.e., a residue is conditionally independent of the residues further than distance D away, provided that the residues at most distance D away are known.) For coiled coils, $p = 7$, and we are assuming $D = 4$. We use an $n \times (p+1) \times (D+1)$ matrix A . $A_{k,l,d}$ contains the maximum “score” for r_1, \dots, r_k given that $L_k = l$ and that the block containing k starts at $k-d$. (If $d = D$, the block containing k may start at $k-D$ or earlier.) The matrix A is never actually stored all at once, we need only $O(p(n+D))$ storage.

For convenience, let

$$Q_{k,l,d} = \Pr[L_k = l | L_{k-1} = \nu^{-1}(l)] \cdot \\ \Pr[\mathcal{R}_k | \mathcal{R}_{k-d} \wedge \cdots \wedge \mathcal{R}_{k-1} \wedge \\ \wedge L_{k-d} = \nu^{-d}(l) \wedge \cdots \wedge L_k = l],$$

where $\nu^{-2}(l) = \nu^{-1}(\nu^{-1}(l))$, etc. Equation 4.6 and the Markov chain transition probabilities are used to compute $Q_{k,l,d}$. Then

$$A_{k,l,d} = \log Q_{k,l,d} + \begin{cases} 0 & k=1, d=0 \\ -\infty & k=1, d>0 \\ A_{k-1, \nu^{-1}(l), d-1} & k>1, 1 \leq d < D \\ \max_{D-1 \leq d' \leq D} A_{k-1, \nu^{-1}(l), d'} & k>1, d=D \\ \max_{0 \leq d' \leq D} \max_{l' \neq \nu^{-1}(l)} A_{k-1, l', d'} & k>1, d=0 \end{cases}$$

The first two cases are for initialization; residue 1 begins its block. The third term is for the case where residue k starts a new block. The last two terms are for the case where residue k continues the block that residue $k-1$ is in. In the $d = D$ case, the two terms correspond to whether residue k is the D th residue in its block, or whether the block starts even earlier.

To compute the scores for $A_{k,*,*}$, only the scores for $A_{k-1,*,*}$ are used, where $*$ is wildcard for any index. As is usual in dynamic programming, once the matrix entries are computed, the optimal labeling is found by following backpointers. Backpointers need only be stored for $A_{*,*,0}$ and $A_{*,*,D}$, since when $1 \leq d < D$, the backpointers for $A_{*,*,d}$ are fixed. Hence we have

THEOREM 4.1. *The dynamic programming algorithm to compute the optimal assignment of motif positions to amino acids runs in $O(p(n+D))$ space and $O(pDn)$ time, excluding the time and space to compute $Q_{k,l,d}$. When p and D are constant, as they are with coiled coils, $Q_{k,l,d}$ can be computed in constant time, and the algorithm runs in linear time and space.*

Acknowledgments

Peter S. Kim has been a driving force behind this project. We are grateful for his extensive guidance and the opportunity to have worked with him and with his lab. Lenore Cowen and Dan Kleitman have been generous with their time and ideas. Thanks to Dan Gusfield and Teo Tonchev for helpful discussions, and to Mona Singh for comments on the draft.

References

- [1] R. B. Altman. Probabilistic structure calculations: A three-dimensional tRNA structure from sequence correlation data. In *International Conference on Intelligent Systems and Molecular Biology*, pages 12–20. June 1993.
- [2] S. Altschul. A protein alignment scoring system sensitive at all evolutionary distances. *J. Mol. Evol.*, 36:290–300, 1993.
- [3] R. B. Ash. *Basic Probability Theory*. John Wiley & Sons, New York, 1970.
- [4] P. Baldi. Hidden Markov models in molecular biology. Technical report, JPL California Institute of Technology, 1993.
- [5] D. Bashford, C. Chothia, and A. M. Lesk. Determinants of a protein fold: unique features of the globin amino acid sequences. *J. Mol. Biol.*, 196:199–216, 1987.
- [6] B. Berger, D. B. Wilson, T. Tonchev, M. Milla, and P. S. Kim. Paircoil: a program for predicting coiled coils using pairwise residue correlations. Submitted for publication, 1994.
- [7] J. U. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.
- [8] S. H. Bryant and C. E. Lawrence. An empirical energy function for threading protein sequence through the folding motif. *PROTEINS: Structure, Function and Genetics*, 16:92–112, 1993.
- [9] C. M. Carr and P. S. Kim. A spring-loaded mechanism for the conformational change of influenza hemagglutinin. *Cell*, 73:823–832, May 1993.
- [10] P. Y. Chou and G. Fasman. Empirical predictions of protein conformation. *Ann. Rev. Biochem.*, 47:251–76, 1978.
- [11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill; The MIT Press, New York, 1990.
- [12] A. Delcher, K. S., H. Goldberg, and B. Hsu. Protein secondary-structure modeling with probabilistic networks. In *International Conference on Intelligent Systems and Molecular Biology*. June 1993.
- [13] J. R. Desjarlais and J. M. Berg. Redesigning the DNA-binding specificity of a zinc finger protein: a database-guided approach. *Proteins: Structure, Function, and Genetics*, 12:101–104, 1992.
- [14] D. Fass and P. S. Kim. Unpublished results, 1994.
- [15] V. A. Fischetti, G. M. Landau, J. P. Schmidt, and P. H. Sellers. Identifying periodic occurrences of a template with applications to protein structure. *Information Processing Letters*, 45(1):11–18, 1993.
- [16] A. Godzik, A. Kolinski, and J. Skolnick. Topology fingerprint approach to the inverse folding problem. *J. Mol. Biol.*, 227:227–238, 1992.
- [17] M. Gribskov, R. Lüthy, and D. Eisenberg. Profile analysis. In *Methods in enzymology*, volume 183, pages 146–159. Academic Press, San Diego, CA, 1990.
- [18] D. Haussler, A. Krogh, S. Mian, and K. Sjolander. Protein modeling using hidden Markov models: Analysis of globins. In *International Conference on Intelligent Systems and Molecular Biology*. June 1993.
- [19] M. Hendlich, P. Lackner, S. Weitckus, H. Floeckner, R. Froschauer, K. Gottsbacher, G. Casari, and M. Sippl. Identification of native protein folds amongst a large number of incorrect models. *J. Mol. Biol.*, 216:167–180, 1990.
- [20] S. Karlin and V. Brendel. Chance and statistical significance in protein and dna sequence analysis. *Science*, 257:39–49, 1992.
- [21] P. S. Kim, 1994. Personal communication.
- [22] T. M. Klingler and D. L. Brutlag. Detection of correlations in tRNA sequences with structural implications. In *International Conference on Intelligent Systems and Molecular Biology*, pages 225–233. June 1993.
- [23] A. Krogh, M. Brown, S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. Technical Report UCSC-CRL-93-32, University of California at Santa Cruz, 1993.
- [24] S. Lifson and C. Sander. Specific recognition in the tertiary structure of β -sheets of proteins. *J. Mol. Biol.*, 139:627–639, 1980.
- [25] A. Lupas, M. van Dyke, and J. Stock. Predicting coiled coils from protein sequences. *Science*, 252:1162–1164, 1991.
- [26] R. Lüthy, J. U. Bowie, and D. Eisenberg. Assessment of protein models with three-dimensional profiles. *Nature*, 356:83–85, 1992.
- [27] R. Lüthy, A. D. McLachlan, and D. Eisenberg. Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. *Proteins: Structure, Function, and Genetics*, 10:229–239, 1991.
- [28] E. K. O'Shea, R. Rutkowski, and P. S. Kim. Evidence that the leucine zipper is a coiled coil. *Science*, 243:538–542, 1989.
- [29] D. A. D. Parry. Coiled-coils in alpha-helix-containing proteins: analysis of the residue types within the heptad repeat and the use of these data in the prediction of coiled-coils in other proteins. *Bioscience Reports*, 2:1017–1024, 1982.
- [30] M. J. Sippl. Calculation of conformational ensembles from potentials of mean force. *J. Mol. Biol.*, 213:859–883, 1990.
- [31] M. J. Sippl and S. Weitckus. Detection of native-like models for amino acid sequences of unknown three-

dimensional structure in a data base of known protein conformations. *Proteins*, 13:258–217, 1992.

- [32] G. von Heijne and C. Blomberg. Some global β -sheet characteristics. *Biopolymers*, 17:2033–2037, 1978.
- [33] M. S. Waterman and M. D. Perlwitz. Line geometries for sequence comparisons. *Bull. of Math. Biol.*, 46:567–577, 1986.

A Estimating the Individual Probabilities

Since the randomly drawn w -long window can occur anywhere in a sequence in Genpept, positional information as to where it occurs can be ignored. Thus, $\Pr[G_i]$ is estimated by the *relative frequency* of residue A_i in Genpept, which is the number of occurrences of this residue in Genpept divided by the total number of residues in Genpept. Similarly, $\Pr[H_i]$ is estimated by the relative pair frequency of residues A_i and A_{i+d} appearing at distance d apart in Genpept. Below we estimate $\Pr[G_i|C_x]$ and $\Pr[H_i|C_x]$, using approximate frequencies in C . $\Pr[C_x]$ is an unknown constant (it is simply the proportional number of w -long sequences in C to all w -long sequences in Genpept).

We want to estimate $\Pr[H_i|C_x]$ for residue pair A and B in positions i and $i+d$ in C . Let $f_i(A, B)$ be the number of times A and B occur in positions i and $i+d$ in the set of T known samples from C , and $n_j = |\{(A, B) \mid f_i(A, B) = j\}|$.

Then a first pass estimation for $\Pr[H_i|C_x]$ could be the relative frequency j/T on the assumption that unknown sequences in C have the same distribution as known sequences in C , since the latter were picked randomly. However, if one thinks about it more carefully, this is not a good approach when the frequencies are 0. For example, the fact that there are no residues A_i in positions i in the known set does not mean that there is a 0 probability of this event occurring; we may have just been unlucky. Furthermore, setting this probability to 0 causes the probability of the entire sequence being in C to be 0. The 0-frequency case is particularly important in this paper, since we have found pair frequencies are often very likely to be 0.

One method of approximating the probabilities is through a Bayesian estimation [3]. It is assumed that nature picks the desired probability $p \in [0, 1]$, e.g. uniformly at random. We sample and guess p . The guess is $E[p \mid \text{sample data}]$. This is one way to guess p . In the limit of infinite data, it will give us the right answer. However, for the amount of data we had, we found it does not work (i.e., does not produce good separation of positive and negative examples). We found a different approach produces better results.

We compute approximate frequencies for residue pair A and B in positions i and $i+d$ in C as follows.

Define

$$P_{j,j+1} = \frac{j \cdot n_j + (j+1) \cdot n_{j+1}}{(n_j + n_{j+1}) \cdot T} \quad (\text{for } j \geq -1).$$

(Note that the case $n_j = n_{j+1} = 0$ is never used.) Then set approximate frequency $P(A, B) = P_j = (P_{j-1,j} + P_{j,j+1})/2$, where $j = f_i(A, B)$. Thus,

$$(A.1) \quad T \cdot P_j = j + \frac{1}{2} \left(\frac{-n_{j-1}}{n_{j-1} + n_j} + \frac{n_{j+1}}{n_j + n_{j+1}} \right).$$

An approximate single frequency for $\Pr[G_i|C_x]$ is computed similarly.

We show in the following claims that P_j differs from j/T , the observed frequency, by at most $1/(2T)$ and $\sum_j P_j = 1$.

CLAIM A.1. $|(T \cdot P_j) - j| \leq 1/2$, for $j \geq 0$.

PROOF: Consider Equation A.1. In the second term, $-1 \leq -n_{j-1}/(n_{j-1} + n_j) \leq 0$, and $0 \leq n_{j+1}/(n_j + n_{j+1}) \leq 1$; therefore, the second term's total value is between $-1/2$ and $1/2$. \square

CLAIM A.2. $\sum_{j=0}^T n_j \cdot P_j = 1$.

PROOF:

$$\begin{aligned} & \sum_{j=0}^T n_j \cdot P_j \\ &= \frac{n_0 P_{0,1}}{2} + \frac{n_1 P_{0,1} + n_1 P_{1,2}}{2} + \frac{n_2 P_{1,2} + n_2 P_{2,3}}{2} + \dots \\ &= \frac{1}{2} \sum_{j=0}^T P_{j,j+1} (n_j + n_{j+1}) \\ &= \frac{1}{2} \sum_{j=0}^T \frac{(j \cdot n_j + (j+1) n_{j+1})(n_j + n_{j+1})}{T(n_j + n_{j+1})} \\ &= \frac{1}{2T} \sum_{j=1}^T (2 \cdot j \cdot n_j) \\ &= \frac{1}{T} \sum_{j=1}^T j \cdot n_j = \frac{1}{T} \sum_{j=0}^T j \cdot n_j = 1 \quad (\text{by def.}) \quad \square \end{aligned}$$

If $P_{0,1} > 2P(A)P(B)$ (where $P(A)$ and $P(B)$ are the single residue frequencies), then we would conclude $P(A, B) > P(A)P(B)$, even though we saw 0 such events. P_0 would then seem like a poor guess for $P(A, B)$. So for this particular pair (A, B) , we decrease $P_{0,1}(A, B)$ to $2P(A)P(B)$, and evenly redistribute the remaining probability mass to the other $P_{0,1}(X, Y)$'s. This process is repeated until no $P_{0,1}(A, B) > 2P(A)P(B)$, or it cannot be applied anymore. Note that the second claim still holds because we are simply redistributing the probability mass. In the first claim, the right-hand-side of the equation can be only slightly more than $1/2$.