COS 597c: Topics in Computational Molecular Biology
Lecture 14: November 10, 1999
Lecturer: Larry Brown
Scribe: Jessica Bessler [1]

The Threading Approach to Tertiary Structure Prediction

Today we will study the problem of predicting the tertiary structure of a given protein sequence. In particular, we will focus on the *threading* or *sequence-structure alignment* approach to this problem.

The threading and sequence-structure alignment approachs are based on the observation that many protein structures in the PDB are very similar. For example, there are many 4-helical bundles, TIM barrels, globins, etc. in the set of solved structures. As a result of this, many scientists have conjectured there are only a limited number of "unique" protein folds in nature. Estimates vary considerably, but some predict that are fewer than 1000 different protein folds. Thus, one approach to the protein structure prediction problem is to try to determine the structure of a new sequence by finding its best "fit" to some fold in a library of structures (see Figure 1).

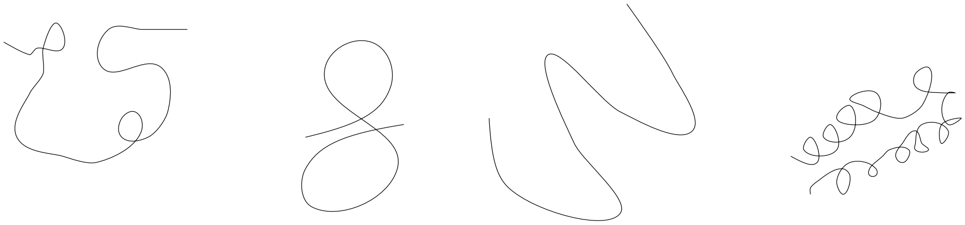

Figure 1: Given a new sequence `MLDVKAYKEMNT...`, and a library of known folds, the goal is to figure out which of the folds (if any) is a good fit to the sequence.

As a subproblem to fold recognition, we must solve the *sequence-structure alignment* problem. Namely, given a solved structure T for a sequence $t_1 t_2 \dots t_n = t$ and a new sequence $s_1 s_2 \dots s_m = s$, we need to find the "best match" between s and T.

This actually consists of two subproblems:

- Evaluating (scoring) a given alignment of s with a structure T.

[1] This portion of lecture 14 is adapted from one given by Mona Singh at MIT. Scribe notes are adapted from notes taken by Valentin Spitovsky also at MIT.

- Efficiently searching over possible alignments.

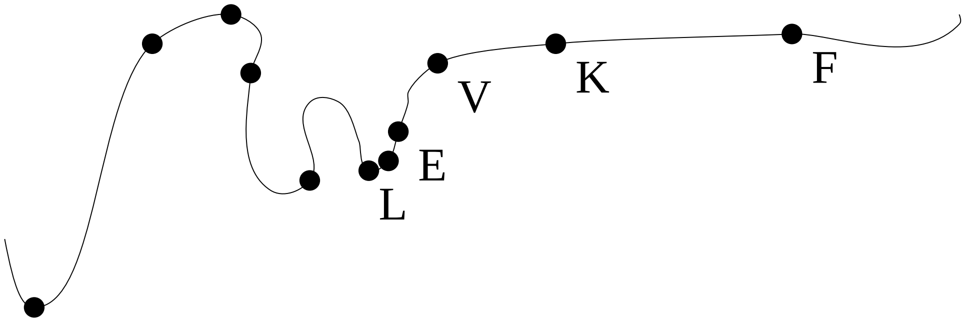

Figure 2: Example: New sequence s=`LEVKF`, and its best alignment to a particular structure.

There are at least three approaches to the sequence-structure alignment problem.

1. The first method is to just use protein sequence alignment. That is, find the best sequence alignment between the new sequence s and the sequence t with structure T. This is then used to infer the structural alignment: if s_i aligns with t_j, s_i's position in the 3D structure is the same as t_j's.

   Scoring in this case is based on amino-acid similarity matrices (e.g., you could use the PAM-250 matrix), and the search algorithm is dynamic programming ($O(nm)$ time).

   This is a non- physical method; that is, it does not use structural information. The major limitation of this method is that similar structures have lots of sequence variability, and thus sequence alignment may not be very helpful. Hidden Markov model techniques have the same problem.

2. The second method we will describe, the 3D profile method, actually uses structural information [1]. The idea here is that instead of aligning a sequence to a sequence, we align a sequence to a string of descriptors that describe the 3D environment of the target structure. That is, for each residue position in the structure, we determine:

   - how buried it is (buried, partly buried or exposed)
   - the fraction of surrounding environment that is polar (polar or apolar)
   - the local secondary structure (α-helix, β-sheet or other)

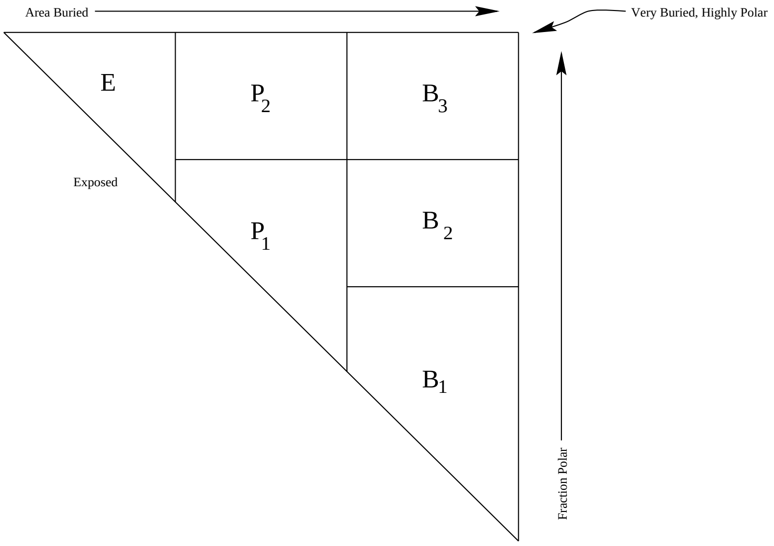

Figure 3: We assign 6 classes of environments to each position in the structure. These environments (E, P_1, P_2, B_1, B_2 and B_3) depend on the number of surrounding polar residues and how buried the position is. Since there are 3 possible secondary structures for each of these, we have a total of $6 \times 3 = 18$ environment classes.

For each position in the structure, we categorize it into one of 18 environment classes using these characteristics (see Figure 3). Because we are using environmental variables, this adds a physical dimension to the problem.

The key observation is that different amino acids prefer different environments. For all proteins in the PDB, we can tabulate the number of times we see a particular residue in a particular environment class, and use this to compute a score for each environment class and each amino acid pair. In particular, we compute a log-odds score of

$$\text{score}_{ij} = \ln\left(\frac{Pr(\text{residue } j \text{ in enviroment } i)}{Pr(\text{residue } j \text{ in any enviroment})}\right)$$

The denominator is obtained from amino acid frequencies present in the PDB

This gives us an 18x20 table as follows:

Environment Classes	W	F	Y	$\cdots$
$B_1\alpha$	1.00	1.32	0.18	$\cdots$
$B_1\beta$	1.17	0.85	0.07	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Then we can build a 3D profile for a particular structure using this table. Namely, for each position in our structure, we determine its environment class, and the score of a particular amino acid in this position depends on the table we built above. Thus, for example, if the first position in our structure has environment class $B_1\beta$, the score of having a tyrosine (Y) in that position is 0.07. Thus, for example, if there are n positions in our structure, we build a table as follows:[2]

Position in Fold	Environment Class	W	F	Y	$\cdots$	Gap Penalty
1	$B_1\beta$	1.17	0.85	0.07	$\cdots$	200
2	E loop	-2.14	-1.90	-0.94	$\cdots$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$			

Then to align a sequence s with a structure, we align the sequence with the descriptors of the 3D environment of the target structure. To find the best alignment, we use a 2D dynamic programming matrix as for regular sequence alignment:

$$\begin{array}{c|l} & e_1 e_2 \cdots e_n \leftarrow \text{environment classes} \\ \hline s_1 & \\ s_2 & \\ \vdots & \\ s_m & \\ \uparrow & \\ \text{new sequence} & \end{array}$$

Thus, to use the 3D profile method for fold recognition, for a particular sequence we calculate its score (using dynamic programming) for all structures. Significance of a score for a particular structure is given by scoring a large sequence database against the structure and calculating

$$z_{-\text{score}} = \frac{\text{score } - \mu}{s}$$

where μ is the mean score for that structure, and s is the standard deviation of the scores.

The advantages of the 3D profile method over regular sequence alignment is that environmental tendencies may be more informative than simple amino acid

[2] The gap penalties are chosen to discourage gaps in positions within α-helices and β-sheets.

similarity, and that structural information is actually used. Additionally, this is a fast method with reasonably good performance. The major disadvantage of this method is that it assumes independence between all positions in the structure.

3. Our third method for sequence-structure alignments uses *contact potentials.* Most "threading" methods today fall into this category.

   Typically, these methods model interactions in a protein structure as a sum over pairwise interactions.

   One formalization of the problem is:

   Given: a structure P with positions $p_1, p_2, \ldots, p_n$, and a sequence $s_1, \ldots, s_m$.
   Find: $t_1, t_2, \ldots, t_n$ (where $1 \leq t_1 < t_2 < \cdots < t_n \leq m$ and t_i indicates the index of the amino acid from s that occupies p_i) such that

   $$\sum_{i=1}^{n} \sum_{j=1}^{n} \text{score } (i, j, s_{t_i}, s_{t_j})$$

   is maximized.

   This problem is NP-complete for pairwise interactions. (If the contact graph for a structure is planar, there are approximation algorithms for this problem. However, in practice, they are not used because most graphs would not be planar and heuristics are thought to give better solutions.) One approach commonly used to find threadings is to disallow gaps into core segments (such helices and sheets), and to put lower and upper bounds on distances between core segments. Some algorithms also use exhaustive enumeration and branch and bound techniques to find the best threading. Alternatively, some approaches give up the guarantee of finding the best threading, and use fast heuristics instead.

   The score functions come from database-derived pairwise potentials. The general idea is to define a cutoff parameter for "contact" (e.g., up to 6 Angstroms), and to use the PDB to count up the number of times amino acids i and j are in contact:

   $$\text{score}_{ij} = \ln \left(\frac{Pr(i, j | \text{ contact })}{\text{normalization}} \right).$$

   There are several methods to do this normalization. For example, in [2], normalization is by expected frequencies.

   Additionally, there are many variations in defining the potentials. For example, in addition to pairwise potentials, some researchers consider single residue potentials as well (e.g., to take into account hydrophobicity or secondary structure), or distance-dependent intervals (e.g., counting up pairwise contacts separately for intervals within 1 Angstrom, between 1 and 2 Angstroms, etc.).

An interesting question for the threading approach is that as the number of known folds increases, will threading methods improve? That is, will better potential functions give better results?

In the next part of the lecture, we will look more closely at the knowledge-based potentials.

References

[1] J. Bowie, R. Luthy and D. Eisenberg. "A method to identify protein sequences that fold into a known three-dimensional structure." *Science*, 253:164-170, 1991.

[2] S. Bryant and C. Lawrence. "An Empirical Energy Function for Threading Protein Sequence Through the Folding Motif." *Proteins: Structure, Function and Genetics*, 16:92–112, 1993.