4
Evolutionary-based Methods for Predicting Protein Structure: Comparative Modeling

4.1
Introduction

Historically, methods for predicting protein structure are distinguished according to the relationship between the target protein(s) and proteins of known structure. "Comparative modeling" is the name given to the set of techniques that can be applied when a clear evolutionary relationship between the target and a protein of known structure can be easily detected from the sequence. "Fold-recognition" is the name given to methods that can be applied when the structure of the target protein turns out to be related to that of a protein of known structure although the relationship is difficult, or impossible, to detect from the sequences. In other words, the target protein can have a very distant relationship with a protein of known structure so that sequence-based methods are not sufficient, by themselves, to ensure they are part of the same family (homologous fold recognition) or share a fold with a known protein for reasons other than evolution (analogous fold recognition). Finally, when neither the sequence nor the structure of the target protein are similar to that of a known protein, we classify the methods as techniques for new fold prediction. The subdivision is somewhat artificial – the distinction between comparative models and homologous fold recognition is based on the ability of sequence-based methods to detect evolutionary relationships and therefore depends on the method used to detect the relationship. The distinction between analogous fold and new fold recognition, on the other hand, relies on our definition of similarity between folds and here also the demarcation line is rather fuzzy.

We will follow the classical subdivision in this book mainly for ease of reference, but the reader should not be led to believe that, in building a model that falls into one of these broad categories, she or he can safely ignore the advances and pitfalls of the others. Rather, each technique is providing information and tools that can be used throughout the range of targets of three-dimensional structure modeling. This obviously implies that one must often refer to aspects and techniques that are discussed in the context of a different

Protein Structure Prediction. Edited by Anna Tramontano

methodology. To simplify matters, Figure 4.1 tries to provide a guide to structure prediction pointing to the relevant sections that contain topics related to each of the steps of the procedure.

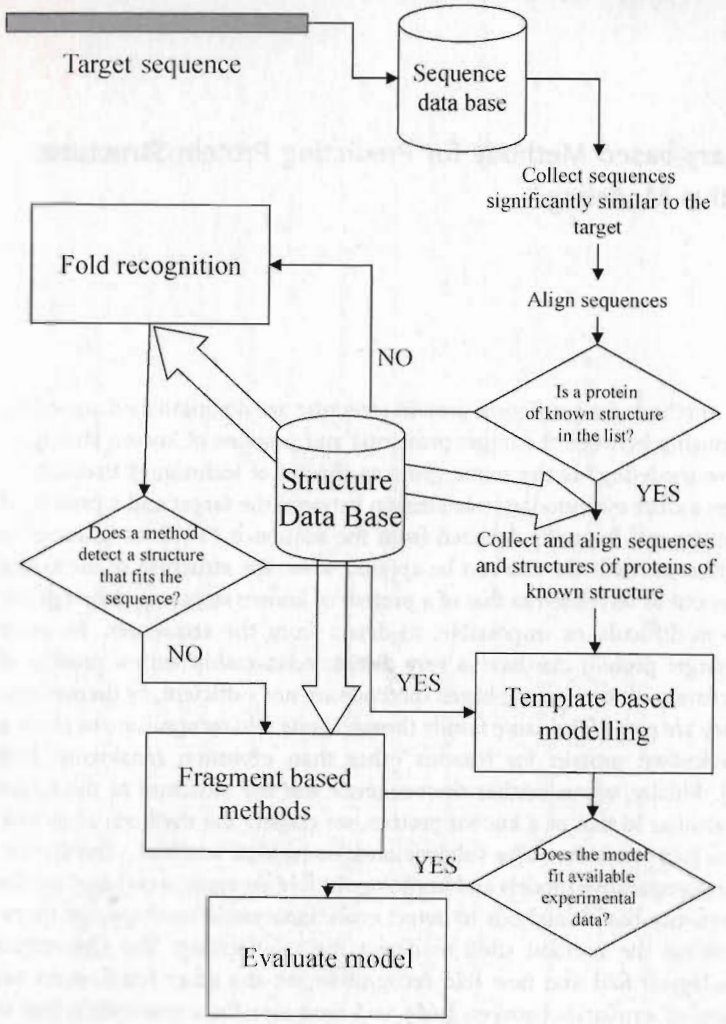


Figure 4.1 A guide to protein-structure prediction. The first step is always a search in the protein sequence database. Comparative modeling should be used when a protein of known structure sharing sequence similarity with the protein under examination is present in the database. If this is not so, fold-recognition methods should be applied and, should they fail, the user should resort to new fold or fragment-based methods. Note the central role played by the structure database in all these heuristic methods.

4.2
Theoretical Basis of Comparative Modeling

Comparative modeling is the most used method for predicting the structure of proteins. There are two reasons for this. First, the quality of models based on

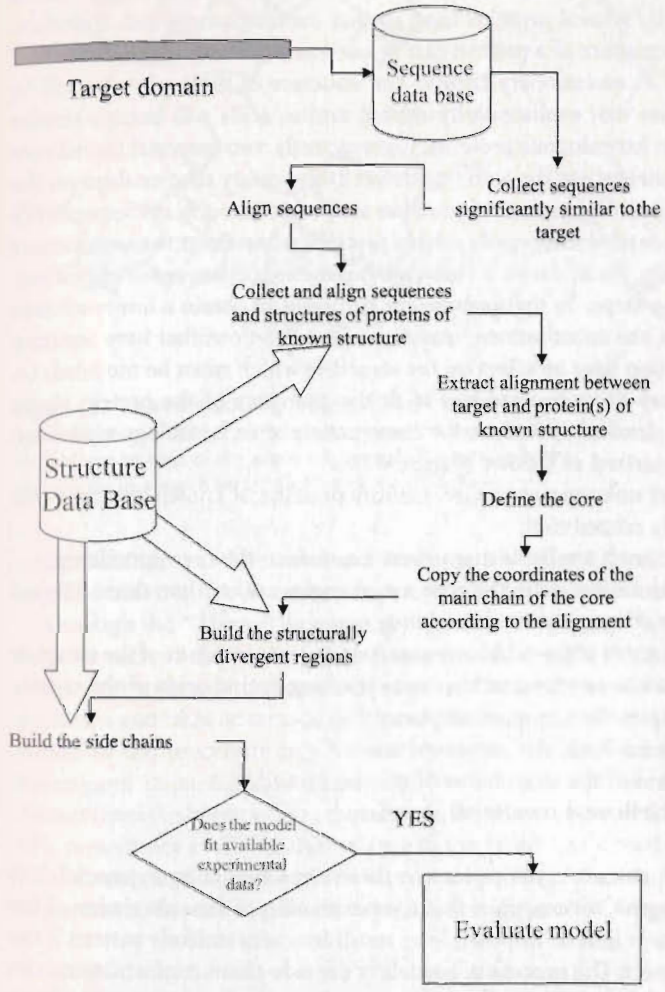


Figure 4.2 Schematic diagram of a typical comparative modeling procedure. The protein of interest should first be split into its domains. For each domain, sequences similar to the target sequences should be collected using a database search tool such as FASTA, BLAST, or PSI-BLAST. The sequences retrieved should be realigned using a multiple sequence alignment program (for example CLUSTAL or T-COFFEE). The implied alignment between the target protein and the protein(s) of known structure will form the basis of construction of the model. This can proceed by first building the main chain of the core regions, then the main chain of the structurally divergent regions, and, finally, the side-chains. The final evaluation of the model should take into account any available information on the protein of interest.

reasonably close evolutionary relationships have been shown to be more accurate, on average, that those produced with different techniques. Second, the expected reliability of the final model can be estimated a *priori*. The latter is not a trivial point, it enables a decision to be made about whether a model can be sufficiently accurate to provide the required answers for the biological problem at hand.

The fundamental concept that forms the basis of comparative modeling methods is that evolutionarily related proteins have similar conformations and, therefore, the experimental structure of a protein can be used as a starting model for that of other members of its evolutionary family. The structure of these proteins will be similar in the sense that evolutionarily related amino acids will occupy similar relative positions in homologous proteins. Consequently, two essential ingredients of comparative modeling are the ability to detect evolutionary relationships on the basis of the amino acid sequence of proteins and to deduce the correspondence between amino acids of evolutionarily related proteins that reflects the evolutionary history of the family, i.e. to derive a biologically meaningful sequence alignment. Although these two steps, by themselves, are sufficient to obtain a low-resolution model of a protein, the substitutions, insertions, and deletions that have accumulated during evolution have an effect on the structure which must be modeled, i.e. the known structure must be modified to fit the sequence of the protein under examination. The classical procedure for construction of an homology model can therefore be summarized as follows (Figure 4.2):

- given a protein of unknown structure, identify proteins of known structure that are evolutionarily related to it;
- if they exist, construct a reliable alignment, i.e. deduce the correspondence between related amino acids in the core, i.e. in regions other than those affected by insertions, deletions, and local refolding;
- assign the coordinates of the backbone atoms of the core residues of the template protein to the backbone atoms of the corresponding amino acids of the target protein according to the sequence alignment;
- model the regions outside the conserved core;
- model the position of the side-chains of the target; and
- optimize the final three-dimensional structure.

There is no reason, at least in principle, why these steps should be sequential. It is not difficult to imagine, for example, that the positioning of the side-chains of the residues of the core might be impossible or result in a very unlikely pattern if the alignment is incorrect. The process of modeling the side-chain conformations can therefore require that the original alignment is modified. Indeed this is not unusual and the alignment must often be modified during the procedure, as illustrated by the example in Figure 4.3.

Ideally, one would like to optimize the final model and not each single step of the process. The task is to find the evolutionarily related structure(s), the alignment, divergent region, and side-chain-building procedure that, taken together, optimize parameters that correlate with the accuracy of the final model. In practice, this cannot be done both because it is computationally prohibitive (too many variables

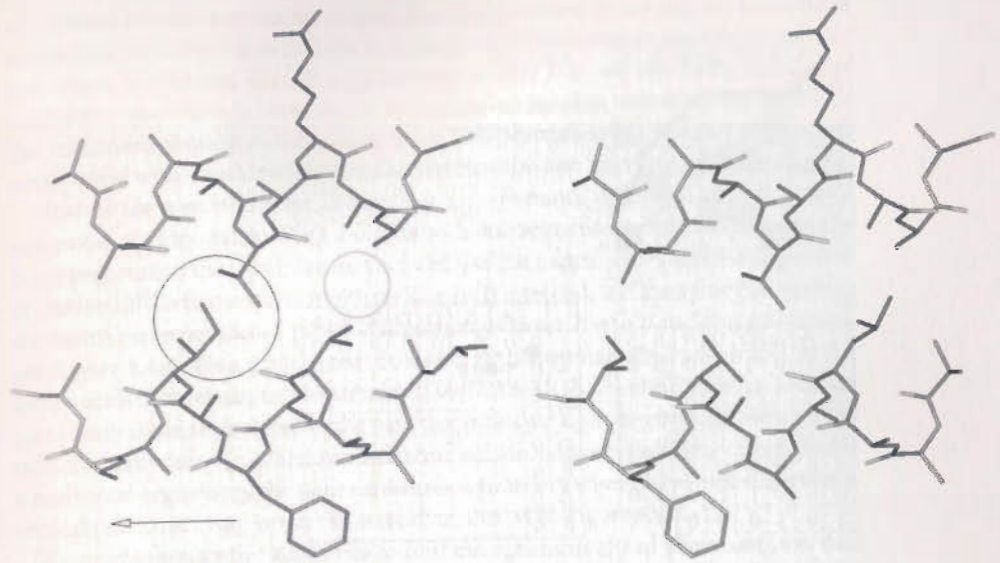


Figure 4.3 The model-building procedure might help refining the initial alignment. In the example shown here it is apparent that a shift of one residue of the sequence towards the carboxy-terminus results in a much better packing of the side-chains.

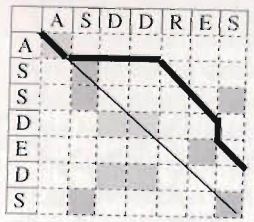
and possibilities) and because, as already mentioned, we do not have a satisfactory way of evaluating the accuracy of a model on the basis of its atomic coordinates.

Although the "classical" sequential procedure can be modified in several ways to improve each of the steps and overcome the limitations of the approach, it is still convenient to briefly discuss each step separately, because each leads to several problems and takes advantage of different techniques and methods. The sequential nature of the procedure implies that errors in one step are bound to affect all subsequent steps. Clearly, selection of the wrong template, i.e. of a protein very distant from the target when other, better, templates are available or – even worse – of a protein not evolutionarily related to the target has devastating effects on the final model; equally serious are errors in sequence alignment – it is extremely important to be very careful especially in the first part of the procedure.

4.3 Detection of Evolutionary Relationships from Sequences

The first question that arises is how to detect an evolutionary relationship between two proteins. Let us assume that the optimum sequence alignment, i.e. that which reflects the evolutionary relationship between two protein sequences, is that which minimizes the differences between them. In this hypothesis, and if we ignore for the moment insertions and deletions, the alignment can be computed exactly with

a)



b)

```
ASDDRES
ASSDEDS

ASDDRE-S--
A---SSDEDS
```

c)

	A	S	D	D	R	E	S
A	1	0	0	0	0	0	0
S	0	1	0	0	0	0	1
S	0	1	0	0	0	0	1
D	0	0	1	1	0	0	0
E	0	0	0	0	0	1	0
D	0	0	1	1	0	0	0
S	0	1	0	0	0	0	1

d)

		A	S	D	D	R	E	S
	0	0	0	0	0	0	0	0
A	0	1	0.4					
S	0							
S	0							
D	0							
E	0							
D	0							
S	0							

e)

		A	S	D	D	R	E	S
	0	0	0	0	0	0	0	0
A	0	1	0.4	0	0	0	0	0
S	0	0.4	2	1.4	0.8	0.2	0	1
S	0	0	1.4	2	1.4	0.8	0.2	1
D	0	0	0.8	2.4	3	2.4	1.8	1.2
E	0	0	0.2	1.8	2.4	3	3.4	2.8
D	0	0	0	1.2	2.8	2.4	3	3.4
S	0	0	1	0.6	2.2	2.8	2.4	4

Figure 4.4 The Needleman and Wunsch alignment algorithm. A path in the matrix corresponds to an alignment. In the example, the thin line in part *a* of the figure corresponds to the first alignment shown in part *b*. The line runs diagonally and therefore corresponds to an alignment where there are no insertions or deletions. The tick line, instead, contains an horizontal line (indicating that the amino acids SDD of the first sequence do not correspond to any amino acid of the second and therefore represent an insertion in the first sequence) and two vertical lines (implying that the amino acid D and the final DS pair of the second sequence do not correspond to any amino acid in the first and is an insertion in the second sequence or, equivalently, a deletion in the first). To compute the optimum alignment we fill the cells of the matrix (part *c*) with a number representing the likelihood that the amino acid in the row is replaced by that in the column. In this example we assign 1 to identical amino acids and 0 to different ones. Part *d* shows the construction of the cumulative matrix as described in the text.

algorithms known as dynamic programming algorithms, as we will see later. Next, we can ask how likely it is that the resulting similarity between the two sequences has arisen by chance. If such a probability is very low, we can infer with some confidence that the two proteins are homologous. In other words we can measure the minimum sequence distance between two proteins and, if such a distance is small, infer with some statistical reliability that the two proteins are homologous.

Most of the algorithms for computing the optimum alignment of two protein sequences start by constructing a matrix with one sequence in the first row and the other sequence in the first column. Each cell *ij* of the matrix represents the alignment of residue *i* of the first sequence with residue *j* of the second, and therefore each path in the matrix corresponds to a possible alignment between the first and second sequences (Figure 4.4 a). We are interested in the path that maximizes the fraction of identical amino acids between the two sequences. If we fill the *ij* cell when the amino acids i and j are identical, we are looking for the path that includes the maximum number of filled cells. The path cannot run backward and can include horizontal and vertical segments: a horizontal segment of the path corresponds to an insertion in the first sequence, a vertical one to an insertion in the second, as shown in Figure 4.4 a.

We might require the algorithm to find the optimum global alignment, i.e. that starting from the first cell in the upper left corner and ending in the last, in the lower right corner, or local ones, i.e. only includes high-scoring segments (segments containing many filled cells) between the two sequences.

Insertions and deletions are rarer events than substitutions, therefore we must tell the algorithm that vertical and horizontal moves must be penalized compared with diagonal moves. It is difficult to correctly estimate the penalty that must be attributed to insertions and deletions. The values used by the different methods are determined heuristically by optimizing the alignment between proteins with known evolutionary relationships, but there is a large body of literature discussing the problem. In general, it is not a good idea to assign the same penalty to each inserted or deleted amino acid. Because insertions and deletions can be more easily accommodated in a protein structure if they occur near the solvent-exposed surface, i.e. in a limited set of positions, it is better to penalize less the continuation of a gap with respect to its initiation.

4.4
The Needleman and Wunsch Algorithm

Figure 4.4 a shows a matrix in which each row corresponds to one character of one string and each column to one character of the other. An element of the matrix is shaded if the characters corresponding to its row and column are identical. A correspondence (alignment) between the two strings is a path in the matrix, as illustrated in Figure 4.4 b. The alignment can be global, i.e., include the whole strings, or local, including only regions of them.

To find the optimal global alignment using our hypotheses, we need the correspondence between the two strings requiring the minimum number of editing

operations, that is in which the number of identical corresponding characters is maximum. If we build the matrix shown in Figure 4.4 c, in which cells corresponding to pairs of identical amino acids are set to unity, we need to find the path that goes from the upper left corner to the lower right corner and includes the maximum number of cells containing "1", i.e. for which the sum of the values of the cells in the path is maximum. Here we are assuming that characters are either identical (scoring 1 in our matrix) or different (scoring 0), but the reasoning can be easily extended to instances in which cells are filled with values reflecting the similarity between the two amino acids of the row and the column, rather than their identity.

Let us build the matrix in Figure 4.4 d, called the cumulative matrix, where in each cell we write the maximum score that can be achieved by any path ending in that cell.

The column and row labeled "0" correspond to inserting or deleting at the beginning of either string. If we do not require the first characters of each string to be aligned, we can set the scores in the first row and first column to 0. Alternatively, we can add a penalty for shifting them, for example, subtracting 0.6 for each shifted character. Calculating the value in the (1, 1) cell is trivial – a path including this cell must include either the cell (0, 0) or the cell (1, 0) or the cell (0, 1). The maximum score achievable by any path ending in (1, 1) is the maximum of:

- 1, i.e. the value in (0, 0) + 1 (because the characters in the first row and the first column are identical and therefore we gain 1 by passing through the (1,1) cell);
- −0.6, i.e. the value in (0, 1) minus the penalty value for an insertion in the horizontal sequence (0.6 in the example); and
- −0.6, i.e. the value in (1, 0) – the penalty value for an insertion in the vertical sequence (0.6 in the example).

We will write this maximum value in the cell (1, 1) and store a pointer to the cell (0, 0) – the cell we used to obtain it.

When (1, 1) is filled, the same strategy can be used to calculate the values of cells (1, 2), (2, 1), and (2, 2), and so on, as shown in Figure 4.4 d. The maximum achievable score for a global alignment of the two strings must be the value in the last cell (7, 7), and this can be obtained if we passed through the cell (6, 6) which was filled using the value in (5, 5), and so on. In other words finding the best path only requires walking backward from (10, 8) to (0, 0) following the pointers.

This algorithm, which is known under the name "Needleman and Wunsch global alignment", can be easily modified to find the best local alignment by starting from the maximum value in a similarly derived cumulative matrix and working our way until we find a 0 (the Smith and Waterman algorithm). In our example these two alignments coincide. The Needleman and Wunsch algorithm guarantees that one optimum path is found. It finds the alignment of two protein sequences that maximizes their identity, or similarity, given a preassigned insertion/deletion penalty.

Question: Is it reasonable to modify the alignment manually taking into account other information?

»In comparative modeling the experimental structure of one of the proteins in the alignment is known and we know that the structure of the other is very similar. The alignment already gives us precious information about the location of the secondary structure in both proteins and about their overall architecture, including which regions are exposed to the solvent and which are buried in the core of the proteins. By inspecting the alignment and the structure of the known protein, it is possible to manually adjust the positions where insertions and deletions are more likely to be located. For example, if the alignment algorithm has positioned a gap inside a secondary structure element, it is advisable to move it to the beginning or end of the element. This is a perfectly legitimate procedure. The alignment algorithm maximizes a predefined score and, in general, has no information about the structural features of the proteins, information that should be taken into account whenever possible.«

4.5 Substitution Matrices

The probability that a substitution is accepted in a protein is not the same for every amino acid change. It is easy to understand that the substitution of a valine with a leucine might be more frequently accepted during evolution than substitution of a glycine with a tryptophan. This can be taken into account by making our alignment matrix more sophisticated – rather than setting to unity the value of cells corresponding to identical amino acids, we can assign to each cell a value that reflects the likelihood that the amino acids in the row and the column are substituted for each other during evolution. These values are reported in substitution matrices, 21×21 tables where the twenty amino acids are in the first row and in the first column and each cell (i,j) contains a value related to the probability that the amino acid i is mutated into amino acid j. Filling the cells corresponding to pairs of identical amino acids and leaving the remainder blank corresponds to an identity substitution matrix, i.e. a substitution matrix where all cells are set to 0 except for the diagonal ones that are set to 1.

Other matrices can be based, for example, on chemical similarity between amino acids or on the minimum number of base substitutions needed to transform a triplet coding for one into one coding for the other. The most commonly used matrices, however, the PAM and BLOSUM matrices (Figure 4.5), are derived from comparisons of evolutionarily related proteins.

PAM (percent accepted mutation) is a unit introduced by Margareth Dayhoff and coworkers to quantify the amount of evolutionary change in a protein sequence. A PAM unit corresponds, on average, to 1% of accepted amino acids changes. PAM1 is a matrix calculated from alignments of sequences at 1 PAM distance from each other. Given these alignments, we compute, for each pair of amino acids i, j, the ratio $f_{ij}/f_i f_j$, where f_{ij} is the frequency with which the two

Pam 250

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Blosum 62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Figure 4.5 The PAM250 (part *a*) and BLOSUM62 (part *b*) substitution matrices. The values corresponding to pairs of amino acids can be used to fill the alignment matrix (part *c* of Figure 4.4).

amino acids are found in corresponding positions in the alignments, i.e. have replaced each other during evolution, and f_i and f_j are the frequencies with which the amino acids i and j occur in the sequences (their product is an estimate of the probability that the two amino acids are found in corresponding positions by chance alone, given the composition of the sequences in the alignment). The ratio $f_{ij}/f_i f_j$ is an estimate of the likelihood that the amino acids i and j are substituted by each other during evolution. Similarity matrices usually report the logarithm to base 2 of these numbers. PAM2 is calculated by multiplying PAM1 by PAM1, PAM3 by multiplying PAM2 by PAM1, and so on (Figure 4.5 a). PAM matrices are based on global alignments of closely related proteins. The higher the number of the matrix, the more suitable it is for aligning distantly related sequences.

The BLOSUM (*bl*ocks *su*bstitution) matrices are instead derived using local alignments of very conserved regions in homologous proteins. They also come as a series of matrices. A BLOSUM-N matrix is derived from alignments such that all sequences sharing more than N% identity with any other sequence in the alignment are averaged and represented as a single sequence (Figure 4.5 b). In contrast with PAM, here a larger number indicates a matrix more suitable for aligning more closely related sequences.

Because substitution matrices, schemes for gap penalty scores, and alignment algorithms are extensively described in many books and articles, we will not go into further detail here and will directly ask the next question – given an alignment score (i.e. the value reported by our alignment algorithm) how likely it is that it reflects a true evolutionary relationship rather than a random similarity of the two amino acid sequences that are, after all, both composed from the same twenty amino acids? In other words, could our alignment score have been obtained when aligning two independent, not evolutionarily related, protein sequences with the same length and composition? If the score obtained by aligning two sequences is significantly higher than the distribution of scores obtained by aligning sets of unrelated sequences, we can say with some confidence that the two aligned sequences are evolutionarily related, but obtaining the "background" distribution of scores for unrelated sequences is not straightforward, because we do not have a validated set of sequences which are definitely unrelated to the ones under examination. One way to approach the problem is to reshuffle our sequences many times, thus obtaining pairs of synthetic sequences with the same composition as our original sequences, but random and therefore not evolutionarily related, align them in pairs and collect the scores for each alignment. The distribution of these scores represents the distribution expected for the scores of alignments between unrelated sequences. We can statistically evaluate whether the score obtained in the real alignment is likely to belong to this random distribution, in which case we assume it is not significant. Indeed programs such as BLAST, PSI-BLAST, and FASTA designed to search a data base for protein sharing a significant sequence similarity to a query protein use this principle to compute the reported probability values, as will be discussed next.

4.6
Template(s) Identification Part I

Step 1:

Given a Protein of Unknown Structure, Identify Proteins of Known Structure that are Evolutionarily Related to it

The task is to use the sequence information on the target protein to detect evolutionary relationships with proteins of known structure. The basic scheme consists in aligning the sequence of the protein of interest with every protein of known structure, compute the sequence identity or similarity and compare the observed value with that expected by chance alone. This is, roughly, what programs such as FASTA or BLAST do – given a "query" protein sequence, they align it with every protein in the data base and report the score of the alignment together with a value related to the probability that the observed similarity is statistically significant. In practice, because of the size of the data base, both programs use approximations to avoid aligning the target sequence with every single sequence. For example, they discard proteins not having at least a short peptide (two or three residues) identical with one of the query protein, compute an approximate score for the alignment of the target with the remaining ones, sort them accordingly, and perform a full fledged alignment only on the subset of sequences likely to be homologous to the query. The details of the approximations used will not be discussed here, both because they are reported in many books, manuals, and articles, and because they are subject to changes in different versions of the tools.

The most relevant issue, as far as the template selection step of comparative modeling is concerned, is evaluation of the score, i.e. how do we decide whether or not a given match is indicative of a structural similarity or, which is equivalent, of an evolutionary relationship? If we are searching a data base to attempt the functional assignment of a protein, we must distinguish between orthologous and paralogous relationships. In modeling this is less of an issue, because both orthologous and paralogous proteins are expected to share a similar structure. We want to make sure that the observed similarity is really indicative of an evolutionary relationship, however.

Both programs use a similar strategy – they compare the observed score between the query protein and each of the database proteins with a random distribution of scores and evaluate how likely it is that the observed similarity belongs to this reference distribution and is, therefore, not significant. With FASTA, the reference distribution of scores is obtained by randomly shuffling the sequence of the query sequence and repeating the search many times on subsets of the original database. BLAST, instead, computes the expected distribution of random scores using as query sequences a set of random sequences with the average composition and length of the database sequences. This implies that BLAST is faster because it does not need to reconstruct the random distribution for each database search, but also that the distribution obtained is based on the assumption that the composition of the query sequence does not differ substantially from the average composition of the proteins in the database. This might not be true – the query sequence can be biased and be

particularly rich in some amino acids, invalidating the statistics. Because of this, the first step in BLAST is to "mask", i.e. not to use in the scoring, those parts of the input sequence that deviate from the composition of the reference distribution. It is important to ensure the masking option of BLAST is turned on; the program output will indicate which regions have been masked and the user can critically evaluate whether they have a relevant effect on the statistics of the results.

Question: Why is it important to take into account the composition of the query protein in evaluating the significance of the score?

»The assumption that the composition of the query sequence is similar to that of the sequences used to derive the random distribution is very important for obtaining meaningful results. Let us assume that our sequence is full of, say, prolines. There will be a high chance that the prolines in our sequence match other prolines in the sequences of the database, even if they are evolutionarily unrelated to the query, and these matches will contribute to increase the score of the alignment. The average score of the random distribution, derived using sequences with a more regular distribution of amino acids, and not containing many matching prolines, will be lower than that expected for the alignment of two random sequences containing many prolines. On average, the random scores will be lower than those obtained by our query sequence with unrelated, but proline-rich, sequences and this might lead us to incorrectly assume that the similarity with the latter is statistically significant.«

How do we measure likelihood in a database search, be it the one from BLAST or FASTA? As we said, we need to compare the score with the expected random distribution and calculate the probability that the observed score belongs to the distribution. For example, if the random distribution were Gaussian, the probability of obtaining a value that is one (three) standard deviation(s) above the mean is

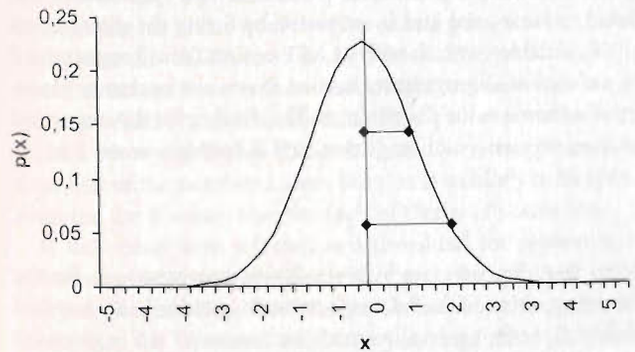


Figure 4.6 A Gaussian distribution with mean = 0 and $\sigma = 1$. The two segments correspond to one and two standard deviations.

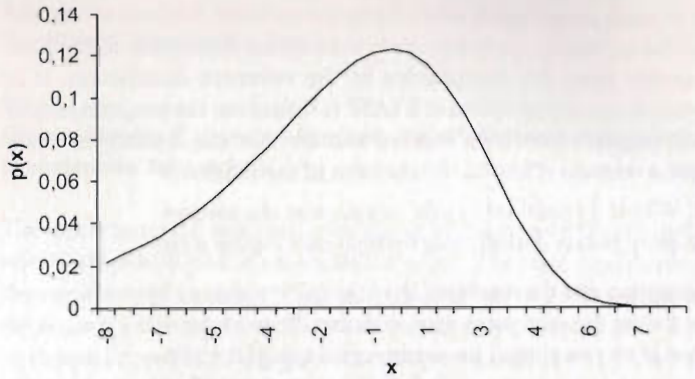


Figure 4.7 Extreme value distribution with $\mu = 0$ and $\beta = 3$. This is the expected distribution for the alignment scores of unrelated sequences.

6% (0.15%). If we obtain a score which is, say, three standard deviations above the mean of the random distribution we can expect to find it by chance alone once every seven hundreds times when aligning unrelated sequences. If we are searching in a database containing two million sequences, we expect to find a few thousand sequences with such a score or higher (Figure 4.6).

The expected distribution of scores of random alignments is not Gaussian. It has been shown computationally that it resembles another type of distribution called "extreme value distribution" (Figure 4.7) often used to model the smallest or largest value among a large set of independent, identically distributed random values representing measurements or observations.

This simply means that the probability of observing a value x, following an extreme value distribution, is:

$$p(x) = 1/\beta \cdot e^{\frac{x-\mu}{\beta}} e^{-e^{\frac{x-\mu}{\beta}}}$$

where β is positive. μ is the location parameter (related to the "position of the distribution" with respect to the x axis) and is estimated by fitting the distribution to the data. The output of a database search with BLAST or FASTA will report the E value, i.e. the expected number of alignments obtaining that score by chance alone, according to the theory of extreme value distribution. The E value for the ungapped alignment of two unrelated sequences of lengths m and n having a score S is:

$$E(S) = Kmne^{-\lambda S}$$

where K and λ are terms that depend on which similarity matrix we use for the amino acids and on the composition of the two sequences. It can also be shown that the cumulative probability of obtaining an alignment with score $S^* \geq S$ is given by:

$$p(S^* \geq S) = 1 - e^{-E(S)}$$

When we compare a query sequence of length m with a database of sequences, we are performing several comparisons with the same m but different n. The approximation used by BLAST is to consider the data base as a single very long sequence and to use for n the value of the total length of the database. Therefore, if the score for the comparison of a query sequence m residues long with one of the database sequences is S^*, the probability that the score is higher than expected by chance is:

$$p = 1 - e^{-KmNS^*}$$

where N is the total length of the database, i.e. the number of residues in the database. This theory is strictly valid for alignments without gaps (insertions and deletions), but it is difficult to prove when gaps are included in alignments; computational simulations have shown, however, that it applies fairly well to gapped alignments also. Notice that the normalized score S' (the bit score) is defined as:

$$S' = (\lambda S - \ln K)/\ln 2$$

The expected value, E, now becomes:

$$E = mn2^{-S'}$$

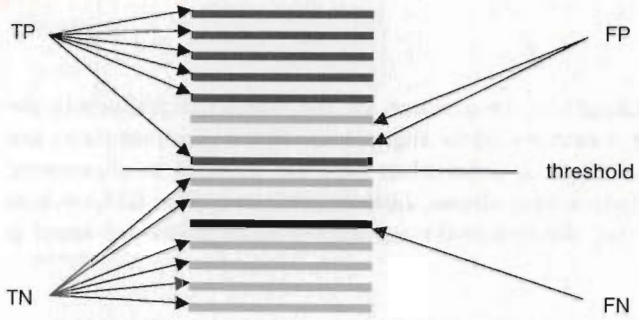
and this can be used to compare the results of different database searches, because it is independent of K and λ.

When we run a database search program, each match reports the score, or the bit score, the E value and the p value. Clearly we are confident in matches for which the E value is very low and the p value very high. When is a match significant or, in other words, what is the right E value or p value threshold that we should use to effectively discriminate between biologically meaningful and random matches? This is one of the most difficult questions in protein bioinformatics, and to understand how to deal with the issue, we need to introduce the concept of sensitivity and specificity. Let us assume we have a protein and a database of sequences some of which are known to be homologous to the query sequence and some known to be unrelated. The set of sequences can be obtained by only including proteins with known three-dimensional structure for which the evolutionary relationship is easier to assess. Now we run a database search and obtain a list of proteins, each with an associated E value. Ideally, we would like all the evolutionarily related proteins to have E values lower than that of the unrelated ones, but this is unlikely to happen. What we can do is to examine the E values and, for each of them, ask ourselves:

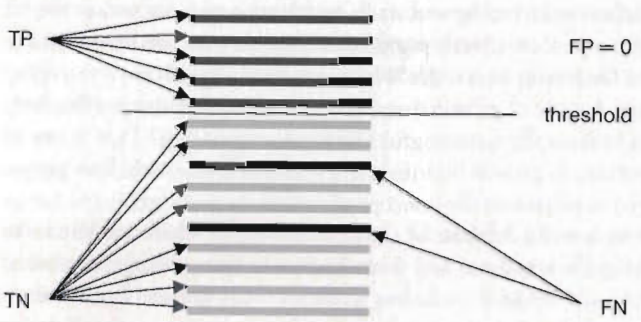
If this value were selected as a threshold for separating related and unrelated proteins, which fraction of the related proteins would be missed and which fraction correctly identified (i.e. how many truly homologous proteins would have an E value lower than the threshold and how many an E value higher than the threshold)?

We call the former "false negatives" (FN) and the latter "true positives" (TP). We then ask ourselves which fraction of unrelated proteins would be mistakenly

labeled as evolutionarily related (with an E value higher than the threshold) and which fraction of unrelated ones would, correctly, have an E value lower than the threshold? These are, respectively the "false positives" (FP) and "true negative" (TN). We define (see Figure 4.8):



Sensitivity = 6/7 = 0.86
Specificity = 6/8 = 0.75



Sensitivity = 5/7 = 0.71
Specificity = 8/8 = 1.00

Figure 4.8 Examples of sensitivity and specificity values for a database search method. In the figure, dark and light segments, respectively, represent proteins homologous and unrelated to the query sequence. If we select the threshold as shown in the top part of the figure, two unrelated sequences will be labeled as "homologous" and one homologous one as "unrelated". A more stringent threshold (bottom), will eliminate false positives, but will increase the number of false negatives.

$$\text{Sensitivity} = TP/(TP + FN)$$

$$\text{Specificity} = TN/(TN + FP)$$

It is important to understand the meaning of these two values and, as always when trying to grasp the meaning of an equation, it is convenient to look at extreme cases. If our E value is very high and we accept every match as significant, we would not miss any TP, no related sequences would escape our attention, but we would also assume that every sequence in the database is related to the query, therefore we would have TN = 0, FN = 0, and, therefore, Sensitivity = 1, Specificity = 0. Our selection would be very sensitive, i.e. not miss anything, but very aspecific, because it would not specifically distinguish between related and unrelated sequences. In the opposite case, where our E value threshold is set to a very low value and every alignment turns out to have an E value higher than the threshold, we would have TP = 0, FP = 0, and Sensitivity = 0 and Specificity = 1. In other words we would never mistakenly assume that two sequences are evolutionary related, but we would not detect most of the true relationships. In summary, a less stringent threshold will increase the fraction of correctly identified true positive, but will also increase the fraction of false positives. If we plot these two values, the true positive and the false positive fraction, against each other, we obtain what is called a ROC (receiving operator curve). The details of the curve clearly depend on the choice of the query sequences, the database, and the settings. An example of a set of ROC curves, with different stringency values, is shown in Figure 4.9.

The FASTA and BLAST strategies are not the only options available for database searching. At least two other general methods are commonly used and, often, are essential for detecting evolutionary relationships – these are profile-based methods and hidden Markov models. They will be discussed after the section describing multiple sequence alignments.

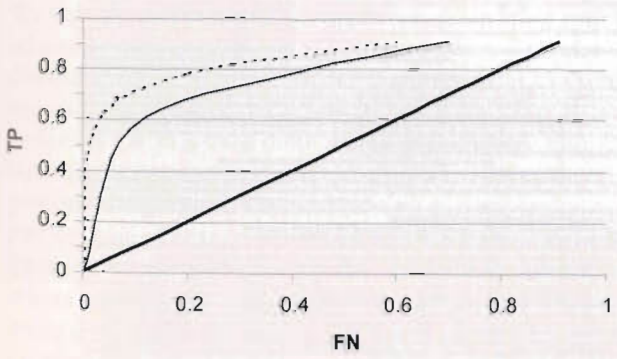


Figure 4.9 Examples of ROC curves. The tick line corresponds to a worthless method, unable to discriminate between positives and negatives. The method represented by the dotted curve is better than that represented by the continuous line: it detects more true positives when finding the same number of false negatives.

4.7
The Problem of Domains

One problem often encountered in database searches is that the query protein might be formed by domains and the detected similarity with a protein sequence in the database can be limited to one of the domains. As already discussed, the significance of the score depends on the length of the query sequence, and assumes that the detected match spans the whole sequence. The database search should be performed separately for each domain of the target protein, but the problem of how to detect the boundary of the domains in a protein from its sequence is still open and there is no clear solution. We can only provide a few practical suggestions.

The size of domains in proteins is usually in the range of 100–200 residues; if, therefore, the sequence of the query protein is much longer than this, it is very likely that the protein is multi-domain. We can try and use one of the methods developed to detect protein domain boundaries. Some are based simply on the expected size of a domain; some take into account the sequence of amino acids, trying to detect linkers between domains; yet others are based on more sophisticated techniques. Although none can guarantee perfect accuracy, their combination can help with the decision whether the sequence must be split, and approximately where.

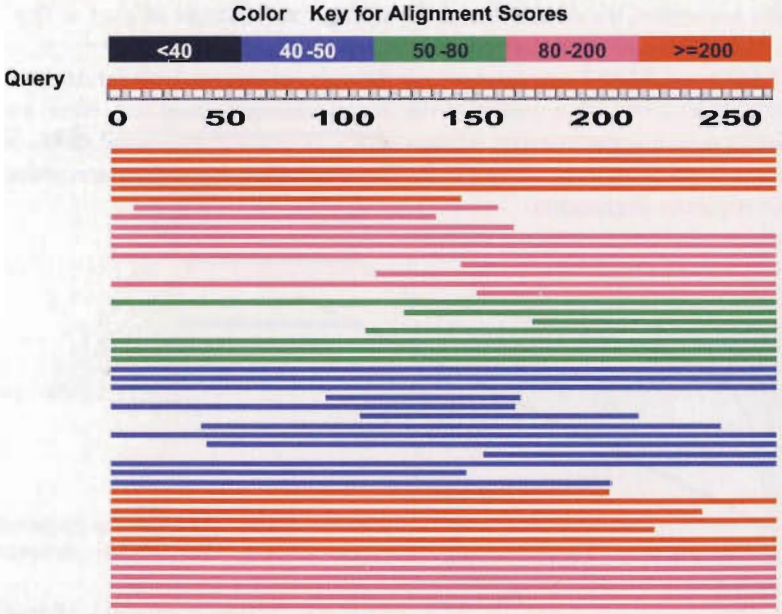


Figure 4.10 Example of the graphical output of BLAST. The example shown suggests that the query protein is formed by two domains, one spanning from the beginning to approximately residue 150, the other from approximately residue 150 to the end of the protein.

Another strategy is simply to split a large sequence into overlapping fragments of the size expected for a domain. We can run our database search using fragment 1–200, then fragment 50–250 and so on.

If the sequence is not extremely long, we can run a first database search using the complete sequence. If a limited region of our protein sequence matches a complete protein, the region probably corresponds to a domain (Figure 4.10) and we should repeat the search only using the sequence of the putative domain. The latter is important in obtaining a significant score. Next, we should search the database with the remaining part of the sequence. A significant match in this latter region might not have shown up in the first search, because both the E and p values depend upon the length of the match.

Finally, it is always a good idea, when we find a match, to "invert the search". In other words, if, in a database search, a significant match is found between protein A and protein B, we should now repeat the search using protein B as a query and verify that protein A is found with a similarly significant score. This procedure protects us from false positive matches arising as a result of peculiar characteristics of our query sequence. If the inverted search does not find the original sequence, it is likely that our search results are not biologically significant and more careful inspection of the properties of the query protein is required.

4.8
Alignment

Step 2:

Construct a Reliable Alignment of the core, i.e. Deduce the Correspondence Between Related Amino Acids in Regions Other than Those Affected by Insertions, Deletions and Local Refolding

We have already outlined the algorithm, the scoring system, and the gap penalty schemes used in alignment, but there are a few more aspects that must be discussed. First, let us recall that any alignment algorithm maximizes the conservation or similarity of paired amino acids. This is because we assume that evolution has preserved those amino acids that are essential for the function and structure of the protein and, therefore, their conservation can guide us through the process of alignment. The conservation of some amino acids might, however, be just because of chance and, when comparing two sequences, we have no way of knowing which pairing of similar amino acids should be weighted more than others because they reflect a genuine evolutionary relationship. One way to overcome this problem, at least partially, is to resort to multiple sequence alignments, i.e. to align together as many sequences as possible of proteins of the family rather than just the target and template sequences. In this way, the amino acids conserved by chance will be different for each pair of sequences, whereas those conserved because of a structural and/or functional constraint will be common to the whole family, as shown in Figure 4.11.

```
Prot1 ILSILHTYSSLNHVYKCQNK.EQFVEVMASALTGYLHTIS..SENLLDAVYSFCLMNYFPLAPFNQLLQKDII
Prot2 IVSILHVYSSLNHVHKIHN..REFLEALASALTGCLHHIS..SESLLNAVHSFCMMNYFPLAPINQLIKENII
Prot3 ISALMEPFGKLNYL..PPNA.SALFRKLENVLFTHFNYFP..PKSLLKLLHSCSLNECHPVNFLAKIFKPLFL
Prot4 IAELIEPFGKLNYV..PPNA.PALFRKVENVLCARLHHFP..PKMLLRLLHSCALIERHPVNFMSKLFSPFFL
Prot5 VQKLVLPFGRLNYL..PLE..QQFMPCLERILARE.AGVA..PLATVNILMSLCQLRCLPFRALHFVFSPGFI
Prot6 VAKILWSFGTLNYK..PPNA.EEFYSSLINEIHRKMPEFNQYPEHLPTCLLGLAFSEYFPVELIDFALSPGFV
Prot7 IPAIIRPFSVLNYD..PPQR.DEFLGTCVQHLNSYLGILD..PFILVFLGFSLATLEYFPEDLLKAIFNIKFL
Prot8 VCSVLLAFARLNFH..PEQEEDQFFSMVHEKLDPVLGSLE..PALQVDLVWALCVLQHVHETELHTVLHPGLH
Prot9 LCSVLLAFARLNFH..PDQE.DQFFSLVHEKLGSEIPGLE..PALQVDLVWALCVLQQAREAELQAVLHPEFH
```

Figure 4.11 A multiple sequence alignment. Note that completely conserved amino acids are easier to detect when more sequences are considered.

This strategy is always beneficial, but it becomes essential when we are attempting to align two distantly related sequences.

The algorithm that we described before for aligning two sequences cannot be extended to many sequences, because it becomes computationally too expensive; multiple alignment methods are, therefore, usually built heuristically. In practice, one first aligns two sequences, then a third to the first two, than a fourth to the first three and so on, with an incremental approach. The alignment algorithm can be extended to align a sequence to an alignment, or an alignment to an alignment (Figure 4.12). Each of the two input sequences or alignments is treated as single sequences but the score at aligned positions is calculated as the average similarity matrix score of all the residues in one alignment relative to all those in the other alignments.

Calculation of the final scoring of a multiple alignment is a more complex problem. One rather unsatisfactory, and yet commonly used, method is to add

Alignment of

PTLRS
LTTRS

with PTLR:

	P L	T T	L T	R R	S S
P	(Score (P,P) + score (L,P))/2	(Score (T,P) + score (T,P))/2	...	...	...
T	(Score (P,T) + score (L,T))/2	(Score (T,T) + score (T,T))/2	...	...	...
L	(Score (P,L) + score (L,L))/2	(Score (T,L) + score (T,L))/2	...	...	...
R	(Score (P,R) + score (L,R))/2	(Score (T,R) + score (T,R))/2	...	...	...

	P L	T T	L T	R R	S S
P	(7-3)/2=2	...	...	...	...
T	(-1-1)/2=-1	...	...	...	...
L	(-3+4)/2=0.5	...	...	...	...
R	(-2-2)/2=-2	...	...	...	...

Figure 4.12 The method for aligning a sequence to an alignment. The alignment is written in the first rows of a matrix and the sequence in the first column. Each cell contains the average between the score of each amino acid of the alignment with the corresponding amino acid of the sequence. The alignment strategy, once the matrix is filled, is identical with that outlined in Figure 4.4.

Sum of pairs score for the alignment :
PTLRS
LTTRS
PTLRT

P L P	T T T	L T L	R R R	S S T
Score (P,L) + Score (P,P) + Score (L,P) = -3+7-3=1	Score (T,T) + Score (T,T) + Score (T,T) = 5+5+5=15	Score (L,T) + Score (L,L) + Score (T,L) = -1+4-1=2	Score (R,R) + Score (R,R) + Score (R,R) = 5+5+5=15	Score (S,S) + Score (S,T) + Score (S,T) = 4+1+1=6
		Score = 39		

Figure 4.13 The score of a multiple alignment can be computed by averaging the scores of each column, as shown in the figure.

the score of each amino acid pair in a column of the multiple alignment and to add the column scores to obtain the alignment score (Figure 4.13). There are several problems with this approach, one of which can be illustrated in Figure 4.14, in which a tree representing the similarity between members of a protein family is shown. Let us consider a multiple alignment including sequences A, B, and C. The sum of scores, for each column, will add the score of the amino acids of A and B, A and C, and B and C. We can think of it as measuring the distance along the path connecting the various nodes, and it is easy to see that the edge indicated by the thicker line in the figure is counted more than once.

Two widely used methods for multiple sequence alignment are CLUSTAL and T-COFFEE. CLUSTAL uses exhaustive pairwise alignments between all the sequen-

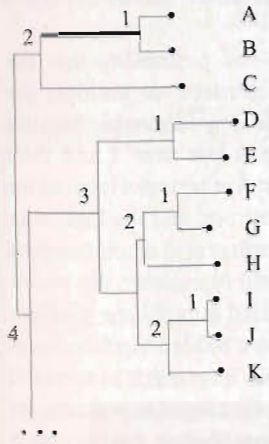


Figure 4.14 A tree constructed on the basis of the sequence similarity among several proteins (indicated by the filled circles). The numbers indicate the order in which the sequences should be iteratively aligned by use of the method described in Figure 4.12, starting from the leaves and proceeding toward the root of the tree.

ces to produce a measure of sequence similarity from which it derives a joining order. This joining order corresponds to a tree that is used to produce the multiple sequence alignment. It should be noted that this tree is not an evolutionary tree. After the joining order has been determined, CLUSTAL aligns pairs of sequences, or pairs of alignments, or one sequence and one alignment starting from the leaves of the tree, so that the most similar sequences are aligned first and most dissimilar ones are added last (Figure 4.14). In CLUSTAL, the matrix used for each alignment is selected according to the sequence distance between the set of sequences to be aligned and the gap penalties depend on the amino acids observed in the column (for example, the presence of hydrophilic or flexible residues in a column reduces the penalty for a gap in that position). The gap penalty is increased for columns that do not contain gaps, if gaps are present nearby in the alignment.

T-COFFEE computes approximate global and local pairwise comparisons of the input sequences and, from these, compiles a list of all pairs of residues observed in at least one of the local alignments, weighted by a factor that depends on the quality of the local alignment around each residue and on how many times that particular pair has been observed. The rationale of this choice is that reliable local alignments are expected to be produced more often by different methods than unreliable ones. These weights, after some more manipulation to ensure consistency, are used as similarity values between the residues for building the multiple sequence alignment.

A multiple sequence alignment can be used to construct profiles, i.e. probability tables that tell us which is the probability of observing each amino acid in each position of a family of proteins. Given a multiple alignment, we can compute the number of occurrences of each amino acid at each position divided by the number of sequences. If the number of sequences is sufficiently high, the resulting table can be seen as a probability table, reporting, for each position, the probability that a given amino acid is present.

Given a newly aligned sequence, we can calculate the probability that the sequence "fits the profile". For each position of the alignment we multiply the probability value corresponding to the amino acid of the new sequence. Because multiplying probabilities is not optimum (they are always less than 1 and their product rapidly becomes very small), the profile is expressed in terms of logarithms of the frequency values. The frequency table can contain zeroes, and the logarithm of 0 is infinite. To avoid this problem, we add 1 to each amino acid count (method of pseudo-counts). With this strategy, profiles can be used to evaluate the probability that a sequence belongs to the family of proteins used to build the profile.

Another, related, but more sophisticated method is to use a hidden Markov model (HMM). This is a way of representing a multiple sequence alignment in terms of "transition" probabilities. We can use an existing multiple alignment to evaluate, for each position, the probability that it is followed by an insertion, a deletion, or a match, and, for the last, the probability it contains each of the twenty amino acids. A hidden Markov model is a representation of the alignment (and therefore of the family of proteins) in probabilistic terms and it can be used to estimate the probability that a new sequence matches the "family model". The very simple HMM shown in Figure 4.15 is derived from the multiple sequence alignment

a)

Multiple sequence alignment:

A C C - E
E C E - A
A C E A A
C - E - E

Counts	Begin-1	1-2	2-3	3-4	4-5	5-end
Match-match	4+1	3+1	3+1	1+1	1+1	4+1
Match-delete	0+1	0+1	0+1	3+1	0+1	0+1
Match-insert	0+1	1+1	0+1	0+1	0+1	0+1
Insert-match	0+1	0+1	0+1	0+1	0+1	0+1
Insert-delete	0+1	0+1	0+1	0+1	0+1	0+1
Insert-insert	0+1	0+1	0+1	0+1	0+1	0+1
Delete-match	0+1	0+1	1+1	0+1	3+1	0+1
Frequencies						
Match-match	0.45	0.36	0.36	0.18	0.18	0.45
Match-delete	0.09	0.09	0.09	0.36	0.09	0.09
Match-insert	0.09	0.18	0.09	0.09	0.09	0.09
Insert-match	0.09	0.09	0.09	0.09	0.09	0.09
Insert-delete	0.09	0.09	0.09	0.09	0.09	0.09
Insert-insert	0.09	0.09	0.09	0.09	0.09	0.09
Delete-match	0.09	0.09	0.18	0.09	0.36	0.09

Counts					
A	2+1	0+1	0+1	1+1	2+1
C	1+1	3+1	1+1	0+1	1+1
E	1+1	0+1	3+1	0+1	2+1
Frequencies					
A	0.43	0.17	0.14	0.5	0.37
C	0.29	0.67	0.28	0.25	0.25
E	0.29	0.17	0.57	0.25	0.37

b)

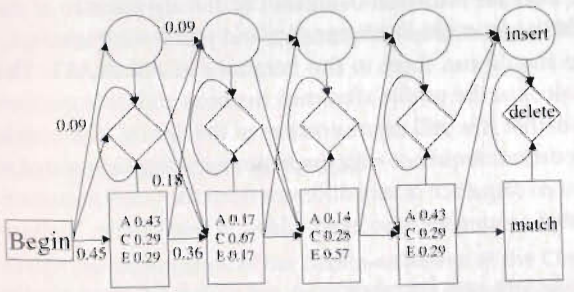


Figure 4.15 Construction of a hidden Markov model. Given a multiple sequence alignment, we first count how many times each transition (Match–match, Match–delete, Match–insert, Insert–match, Insert–delete, Insert–insert, Delete–match) occurs, and add 1 to each. The counts are then transformed into frequencies and used to construct the scheme shown in part *b*. For each match, we also count how many times each amino acid is observed (always adding 1 to the counts) and compute the frequencies. The scheme represents the hidden Markov model of the family of aligned proteins and can be used to calculate the probability that a new sequence belongs to the family, i.e. is generated by the HMM.

shown in the same figure, assuming there exist only three amino acids. In practice, for each position of the alignment we compute the probability that it is followed by a match, an insert, or a delete state, by counting the occurrences of matches, insertions, and deletions in the input alignment (and adding 1 for the pseudo-count). For the match states, we also compute the probability that each of the amino acids is observed in that position. Figure 4.15 shows a commonly used graphical representation of an HMM. If now we have a new sequence, we can compute the probability of the path in the HMM corresponding to it, using algorithms similar to those used for sequence alignments. In this way we can estimate the probability that the new sequence belongs to the family used to construct the HMM.

4.9
Template(s) Identification Part II

Both profiles and HMM can be used to evaluate the probability that a new sequence belongs to the family that generated them. It is only natural to use them to improve the sensitivity of data base searches. A very popular program, called PSI-BLAST is, indeed, based on profile searches. The program first runs a database search using BLAST; it then aligns the sequences that have a significant score, builds a profile, and repeats the search, this time using the profile as query. The new search will, hopefully, identify other sequences belonging to the family that can be aligned to them, so that a new profile can be built and the procedure repeated. In principle, the cycle should be repeated until no new sequences are collected. In practice, there is a significant risk that some false positive proteins slip into the list used to build the profile, making it progressively less specific for the family of interest. This is obviously because a score that appears statistically significant is not necessarily indicative of a true evolutionary relationship, as we have discussed. With PSI-BLAST the problem is aggravated, because statistical evaluation of the significance of the score is much more difficult when multiple alignments and profiles are involved.

As a rule of thumb, one should run three to five iterations of PSI-BLAST. The program allows the user to look at the profile after each iteration and its inspection can help understanding whether it is still representative of the family. The profile can be used to extract an optimum sequence with the highest scoring amino acid at each position. If this optimum sequence is very different from the query sequence, it is very likely that unrelated sequences have been added to our profile, "poisoning" it.

In specific cases, other factors can be taken into account for evaluating the appropriateness of the profile at each iteration step. For example, if proteins of known structure are found among those selected by PSI-BLAST, the user can verify that they have a similar structure and that the alignment of their sequences implied by the profile corresponds to a good structural superposition. If it is known that some amino acids play an essential role in the function or the structure of the family, they should be conserved in each of the sequences and, obviously, included in the optimum sequence.

Although comparison of a new sequence with the HMM of a family is a rather rapid procedure, construction of an HMM is rather computationally expensive. For this reason, one usually compares the query sequence with pre-computed HMM representing different families. These are stored in publicly accessible databases such as PFAM that contains thousands of HMM covering almost three quarters of the known protein sequences.

Question: Can I use the alignment provided by BLAST, FASTA or PSI-BLAST as it is to build a comparative model?

»Although database searching methods have very much improved our ability to detect evolutionary relationships and are continuously being optimized, it is important to understand that the pairwise sequence alignment that they produce is not necessarily the most accurate and that, once a database search has been used to collect the sequences of the family, these should be re-aligned using more accurate methods such as CLUSTAL or T-COFFEE.«

Question: When searching for homologous proteins of known structure, should I only search the database containing the sequences of proteins of known structure?

»Even if we are looking for a similarity between the sequence of our query protein and that of a protein of known structure, we should not limit ourselves to searching the sequences of proteins contained in the PDB database. Proteins of unknown structure are extremely useful for building better profiles and alignments. We should search the entire database, build the alignment of all the members of the family, and only afterwards extract, from this alignment, the alignment of the query sequence with the proteins of known structure.«

4.10
Building the Main Chain of the Core

Step 3:

Assign the Coordinates of the Backbone Atoms of the Core Residues of the Template Protein to the Backbone Atoms of the Corresponding Amino Acids of the Target Protein According to the Sequence Alignment

How do we identify the core or, equivalently, how do we single out those regions that are likely to have preserved their structure during evolution?

If all we have are the sequences of the target and template proteins and the structure of the latter, the problem is complex. Certainly, regions surrounding insertions and deletions have changed their structure somehow, but what about the

others? One can reasonably safely assume that the main elements of secondary structure and residues buried in the protein structure have retained their conformation. But how do we treat exposed loops without insertions and deletions or small domains that are peripheral to the protein structure? Should we assume they are conserved and approximate their structure using the coordinates of the corresponding regions of the template, or try to rebuild them with one of the methods that we will describe later?

Let us recall that for proteins that are closely related, the core is expected to be formed by almost the complete structure, but its extent can decrease rapidly at increasing evolutionary distance (Figure 1.24). If a multiple sequence alignment of the protein family is available, it is usually easier to identify the conserved core. It will correspond to those regions that do not contain insertion or deletions and are well conserved in most of the members of the family. Even more favorable is when we have available the structure of more than one protein of the family. Structural superposition of these proteins will highlight regions that are more prone to change their structure during evolution and these should not be regarded as part of the conserved core. The problem of optimally superposing two or more protein structures does not have a unique solution, but in this case the problem is less severe, because our objective is only to highlight the structurally variable regions and not to obtain precise residue-to-residue correspondence between the proteins.

Another possibility is to see how stable our alignment is, i.e. to rerun the alignment procedure modifying the substitution matrix and, to a limited extent, the gap penalties, to highlight the regions where the alignment is very dependent upon our heuristic variables. When the core has been identified one could use the classical procedure of simply assigning the coordinates of the backbone of the template to the target according to the alignment. In practice, as we will see, more sophisticated methods give better results. In general, when we have at our disposal more than one structure from related proteins we can choose to use different models for different regions, according to the local sequence similarity or take into account all of the homologous structures. The SwissModel server uses this latter approach – after the templates have been identified, their structures are superimposed and the coordinates of well fitting atoms between the various templates, expected to be part of the conserved core, are averaged.

4.11
Building Structurally Divergent Regions

Step 4:
Model the Regions Outside the Conserved Core

The construction of the structurally divergent regions, called SDR, is a serious and open problem in comparative modeling, not only for large regions that have undergone local refolding during evolution, but also for relatively short regions where insertions and deletions occur. It is customary to call these locally refolded

regions "loops", even if they do not necessarily correspond to loops in the protein structure.

First, very careful inspection of the alignment, and of the template structure, is a very important part of the procedure, because the first thing that must be done is to correctly position the gaps and one often has to modify the alignment manually and shift the gaps by a few positions to place them outside regions of secondary structure or of the packed core.

To build the SDR, we can rely on their sequence pattern, on the structure of the corresponding regions in other homologous structures, on the (albeit approximate) knowledge of the regions surrounding them that have been built by comparative modeling techniques, or on energetic calculations. We will discuss these approaches, although it is known that none is currently satisfactory. The problem is not only that the results of these procedures are often incorrect but, more importantly, that it is difficult, if not impossible, to have an a priori estimate of their accuracy. In other words, there are instances when they work and instances when they do not, but it is almost always impossible to tell which beforehand.

Local refolding usually occurs at the periphery of protein structures, in regions that are not subject to a very stringent evolutionary pressure and are, therefore, rarely involved in function, with some notable exception that we will discuss separately. This is good news, because even if we fail to model these regions correctly, it is likely that this will affect to a limited extent only our ability to interpret the information given by a model in terms of its biological significance. It is, nevertheless, a rather frustrating aspect of protein-structure prediction, and probably that which is most embarrassing to modelers.

If a loop is short (three-four residues) we can take advantage of the observed sequence pattern, compare it with the data reported in Table 1.1, and try to model the dihedral angles of its amino acids. As already mentioned, especially if the loop connects two antiparallel beta strands, i.e. if it is a hairpin, there is a correlation between the position of glycine amino acids and the conformation of the loop. This is not as useful as may seem at first sight. First, it is rare that we find insertions and deletions in these tight loops – their structural requirements are such that they are often conserved and rarely involved in local refolding. The other aspect is that, sometimes, tertiary interactions can overcome the sequence requirements and we will see an example of this phenomenon in the section dedicated to immunoglobulin loops.

When loops are medium-sized, it is much more difficult to define them according to their main chain dihedral angles, and consequently the classification becomes less rigorous. It is, however, possible to derive some approximate rules for the structure of these loops, on the basis of the types of interaction that stabilize them. For loops that form compact substructures the main factor determining conformation is the formation of hydrogen-bonds to main chain atoms of the loop. For loops with more extended conformation the required stabilization is obtained by packing an inward pointing hydrophobic side-chain of the loop between the secondary structure elements connected by the loop. The interesting question that arises is, of course, how conserved are such interactions and whether the stabiliz-

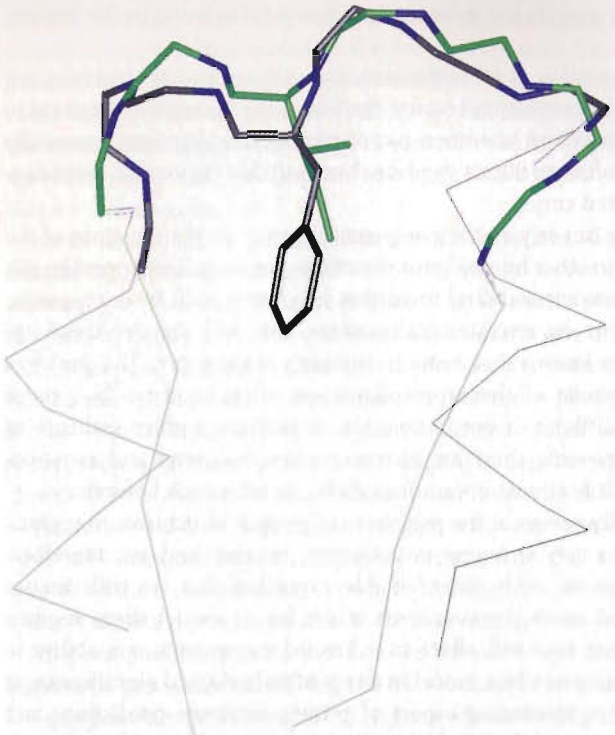


Figure 4.16 The figure shows two loops with similar conformations stabilized by the packing of a central hydrophobic amino acid. Note that one of the loops connects two alpha helices and the other two beta strands.

ing elements can be detected and used to predict the structure of the loop. Figure 4.16 shows one of the many examples that can be used to convince the reader that this is not so.

The structures of two very similar medium sized loops are shown in Figure 4.16. The sequences of the two loops are not similar. In both examples the loop is stabilized by a large and hydrophobic central side-chain which is packed into a cavity of the protein. The first loop connects two strands, however, and the second connects two helices, so it would be impossible to infer their structural similarity from either their local sequence or their context. The other example is shown in Figure 4.17 where there are three very similar loops without any clear local sequence similarity. Two of the three loops have a *cis* proline in an equivalent position and all are stabilized by hydrogen-bonds; such hydrogen-bonds are formed by the residues of the loop with completely unrelated partners, however. In the first loop the partner for these interactions is the side-chain of the residue preceding the loop; in the second it is the main chain of an alanine distant in the primary structure, in the third the propionyl group of a heme. In all these examples the hydrogen-bond partners occupy the same position in space relative to the loop. The structural context of these

three loops is once again completely different. In the first two examples the loop is a hairpin, in the third it connects strands from different sheets.

The conclusion that can be derived from these examples is that the conformation of the loop dictates the interactions required to stabilize it, but in different proteins a variety of different topologies can be used to provide these interactions. This implies that it is unlikely that rules relating sequence to structure can be identified in medium sized loops.

When more than one homologous structure is available, the sequence and length of the region that we are modeling might be more similar to the corresponding region of a protein other than the best template and it is advisable in these instances to use the alternate structure as a local template.

As we have already mentioned, we have available the model of the regions surrounding a loop, usually called stems, and we can try to make use of the knowledge of their structure to model the intervening region. This approach is rather dated, but it still used and sometimes even successful. It originated from a method developed by Alwyn Jones from Uppsala University aimed at building fragments of proteins into electron density maps obtained by X-ray crystallography

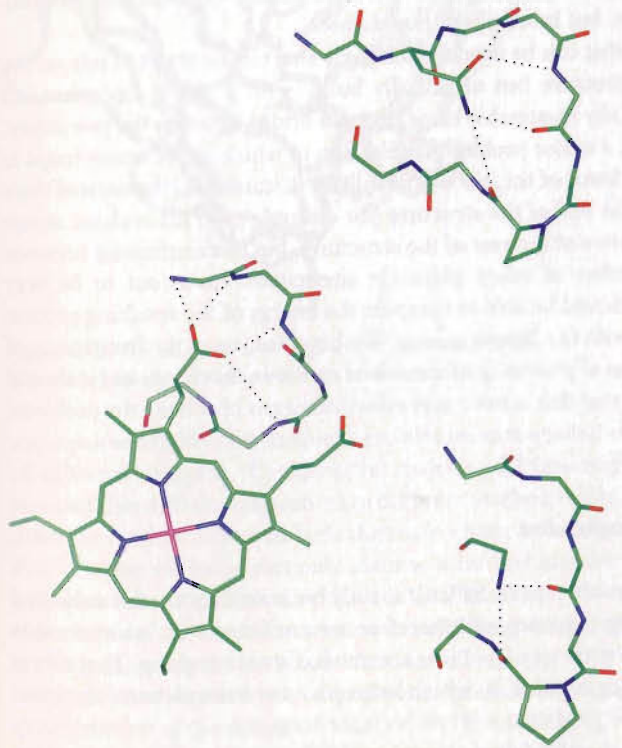


Figure 4.17 The three loops shown in the figure are very similar and stabilized by hydrogen-bonds, however the partners of these interactions are different in the three different proteins (an immunoglobulin, a viral protein, and a cytochrome).

experiments. In the original application of the method one searches in the database of proteins of known structure for regions that fit reasonably well the local electron density in the X-ray map. Clearly, the approach is more useful for loop regions, because regular elements of secondary structure are easy to build manually. The idea that stemmed from this approach is: given the amino and carboxy-terminal ends of a loop, how many ways are there for the amino acid chain to bridge between them? If their number is limited, we might hope that the correct one has already been observed in a known protein structure and, therefore, we can look for regions of known structure that are similar to the stems of the loop to be modeled and contain the correct number of amino acids. If, as almost always happens, more than one is found, we can use the sequence pattern or some energetic consideration to select the most likely.

For this method to work, two conditions should be verified – a loop similar to that to be modeled should be present in one of the proteins of the database of known structure and, if two loops are similar, their stems should also be similar, so we can use the structure of the latter to identify the former. While the first condition is very often verified, the second is not, as shown, for example, by the example depicted in Figure 4.16. Sometimes similar loops have similar stems, especially in evolutionarily related proteins, but more often they do not.

Another approach, that can be used for relatively short loops is not to rely on the database of known structure but to actually build, with a clever algorithm, all possible stereochemically reasonable loops that can bridge between the two stems. With either approach, a major problem is selection of which of the many loops is the most appropriate. Some of the hits can usually be discarded on the basis of their incompatibility with the rest of the structure (for example their main-chain atoms overlap main-chain atoms of the rest of the structure), but discriminating between the usually high number of other plausible alternatives turns out to be very difficult. Ideally, one should be able to compute the energy of the resulting protein and select the model with the lowest energy. We have discussed the limitations of energy-based evaluation of protein conformations earlier in this book, and it should not come as a surprise that this is not a very effective means of solving the problem.

4.12
A Special Case: Immunoglobulins

Predicting the structure of loops is difficult mainly because they are not subjected to a strong evolutionary pressure and therefore we are faced with an enormous number of possibilities with very few hints about what we are seeking. That this is so is proven by immunoglobulins, in which loops play a very important functional role and, indeed, can be predicted with respectable accuracy.

Immunoglobulins are multi-domain proteins consisting of two identical copies of a light, (L) and a heavy (H) chain, each including a variable domain. The antigen binding site is formed by six loops (denoted L1, L2, L3, H1, H2, and H3), clustering in space to form the antigen binding site as shown in Figure 4.18. The high

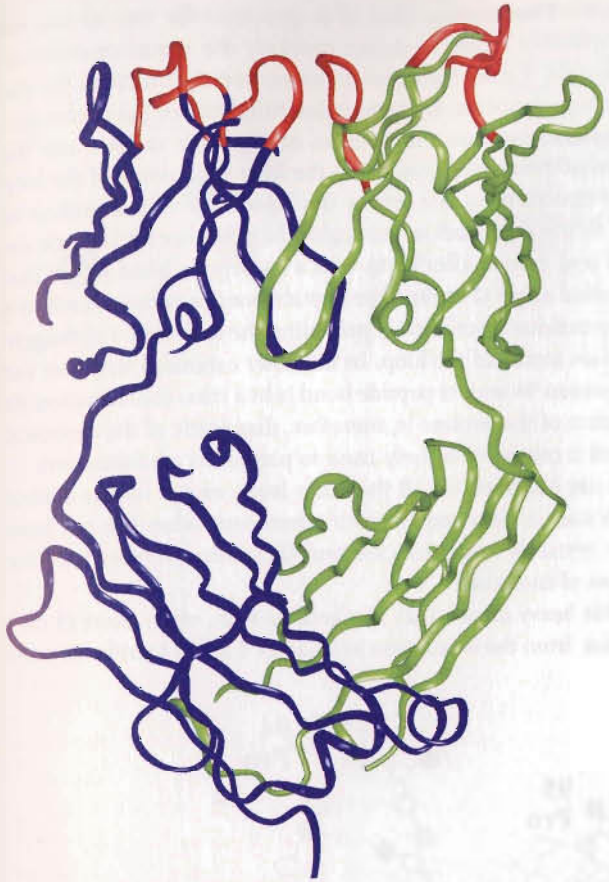


Figure 4.18 The structure of a fragment of an immunoglobulin. The antigen binding loops are shown in red.

sequence variability of these loops enables immunoglobulins to recognize a variety of different antigens. Comparative analysis of immunoglobulin structures has revealed that different sequences in different antibody loops do not always generate different conformations in both the main chain and side-chains of these regions. Five of these six loops can only assume a limited number of main-chain conformations, called "canonical structures". Most sequence variations only affect the side-chains of the loops, consequently modifying the antigen binding-site surface, without changing the backbone structure of the loop. Only some specific sequence changes, in a limited set of positions, produce a change in the main chain conformation of the loops.

Careful analysis of the available structures highlighted the special relationship between sequence changes and canonical structures. There are specific residues which, as a result of their packing, hydrogen-bonding, or ability to assume unusual values of their main chain dihedral angles, are responsible for the occurrence of

each canonical structure. This implies that it is possible, for five of the six immunoglobulin hypervariable loops, to define precisely the sequence-structure relationship. As an example, Figure 4.19 shows the canonical structures for six-residue L3 loops of immunoglobulins. Important determinants for the occurrence of one or the other conformation are the position of a proline residue and the nature of the interaction of the residue preceding the loop with atoms of the loop itself. In the canonical structure on the left in the figure, there is a proline in position 95 (according to the common immunoglobulin numbering scheme devised by Elvin A. Kabat and named after him) with a *cis* peptide bond (recall that this means that the dihedral angle Ω around the peptide bond is approximately 0°) and the side-chain of the residue immediately preceding the loop forms hydrogen-bonds with the main chain atoms of the loop. In the other canonical structure the proline is, instead, in position 94 and its peptide bond is in a *trans* conformation (Ω close to 180°). The position of the proline is, therefore, diagnostic of the canonical structure of the loop and it can be effectively used to predict its conformation.

Analogous situations are observed for all the other loops except for the central region of H3, for which such a clear sequence structure correlation has not been found. These loops are certainly a special case and their structural features are quite unique to this class of molecule.

The second loop of the heavy chain (H2) is a beta hairpin, often short (3 or 4 residues). The expectation, from the discussion in Chapter 1 about hairpin loops, is

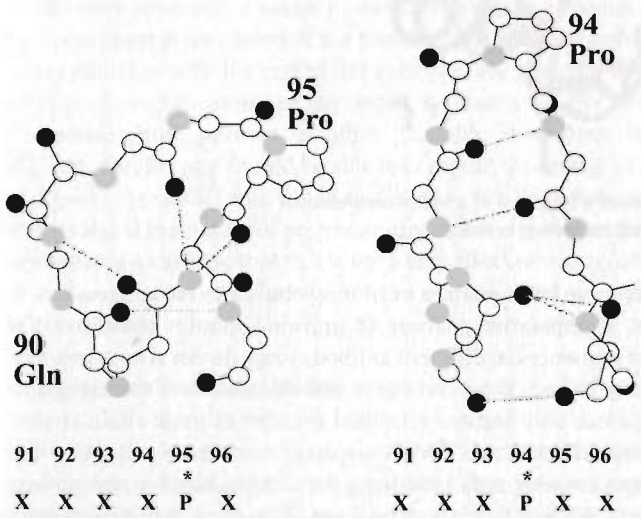


Figure 4.19 The canonical structures of immunoglobulins. The loop shown in the figure is called L3 (it is the third loop of the light (L) chain of antibodies and is part of the antigen-binding site). When the length of the loop is six amino acids, as in the figure, only two main chin conformations are observed. The one on the left occurs when the amino acid in position 95 is a proline. The conformation shown on the right instead occurs when the proline is in position 94. All other residues are free to vary and contribute to the shape of the antigen binding region.

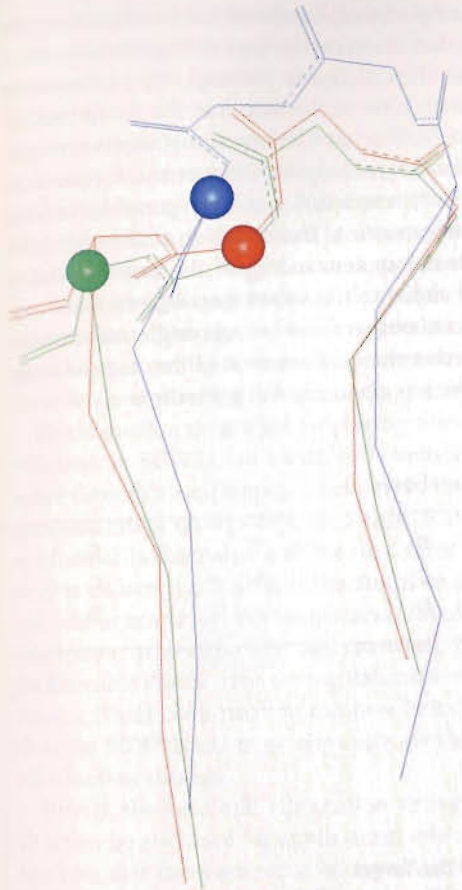


Figure 4.20 Superposition of the H2 hairpin loops of three immunoglobulins. Their conformation does not follow the rules relating sequence and structure in hairpin loops. The determinant of their conformation is the type of amino acid that occupies position 71, and not the position of the glycine (indicated by the sphere in the figure).

that for these short loops there is a correlation between their sequence pattern and their conformation.

An interesting result was obtained by comparing H2 loops of different immunoglobulins of known structure, for example those of 2FBJ, 1NCD, and 2FB4.

All these form a four-residue hairpin turn; their sequence is shown in Table 4.1. If one observes the positions of the glycine units and recalls the discussion in Chapter 1, the conclusion should be that the conformation of the loop of 1NCD and that of 2FBJ should be very similar in that both have a glycine in the fourth position, and different from that of 2FB4, where the glycine is in the second position. What is instead observed is that the conformation of the 2FBJ loop is much more similar to that of the corresponding 2FB4 loop. Careful analysis of the interactions of this loop with the rest of the immunoglobulin structure shows that the determinants of the conformation of this loop involve tertiary interactions, in particular they depend on the size of residue 71, a residue far away in the sequence from the loop, and part of the conserved immunoglobulin conserved framework. When position 71 contains a

small or medium sized residue, for example leucine, as in 1NCD, the conformation of four-residue H2 loops is similar to that illustrated in blue in Figure 4.20; when residue 71 is arginine, as in 2FBJ and 2FB4, different packing of the side-chains arises in the loop region and the main chain of the loop has the conformation illustrated in red and green in Figure 4.20. The implications of this observation are many and relevant both to our understanding of loop architecture and for practical purposes. First, as already stated, a tertiary interaction can be responsible for loop conformation, so our ability to predict the structural conformation of short hairpins may very well be impaired by our limited understanding of the overall stability requirements of proteins. Second, the ability to transplant loop regions from one protein to another, for example from antibodies of non-human origin into human frameworks, relies on the assumption that the conformation of the loops is independent of the rest of the structure, which is obviously not generally true.

Table 4.1 Sequence of the H2 loop of three antibodies.

1NCD	T	N	T	G
2FBJ	P	D	S	G
2FB4	D	G	S	D

4.13
Side-chains

Step 5:
Model the Position of the Side-chains of the Target

Side-chains interact with each other and the energy contribution of their interactions is an important aspect of the stabilization of the native conformation of a protein. The problem of finding the correct combination of dihedral angles of the side-chains, however, is combinatorial in nature. We should inspect every possible combination of side-chain dihedral angles and select the optimum one by optimizing a target function that represents their interaction energy. The energy of the conformation of a side-chain depends upon its tertiary interactions with the rest of the protein structure. However, as originally noted in 1987 by Ponder and Richards, some amino acids have preferences for specific side-chain conformations. The frequencies of the preferred conformations for each amino acid are reported in tables called rotamer libraries. Backbone-independent libraries report the frequencies of each amino acid side-chain conformation, irrespective of the values of their main chain angles. In backbone-dependent libraries the rotamer frequencies are computed as a function of the main chain dihedral angles of the amino acids. Most methods for predicting the side-chain conformations use the conformation with highest frequency for each amino acid as initial values and subsequently modify

them to optimize the energy of their combination. Because conserved side-chains tend to retain their conformation in homologous proteins, their side-chain angles are usually taken from the template, rather than from the rotamer libraries.

The search for the optimum conformation of the side-chains in the model requires a search strategy and a target function to be optimized. The target function is usually an energy function, sometimes simply taking into account steric interactions or using knowledge-based potentials. More specific to the problem of side-chain prediction is the search strategy, which needs to be fast and efficient. The search method can be exact or approximate. Only when it is exact is there a guarantee that the optimum of the target function, for example the knowledge-based potential energy, is found. Approximate methods do not necessarily find the global optimum, but they can be clever enough to reach conformations reasonably close to the one with minimum energy.

The algorithm most used for placing side-chains in a model, and one of the most efficient, is SCWRL. In short, this method first positions each side-chain in its most favorable conformer, unless this is sterically incompatible with part of the modeled main chain other than that of the amino acid itself. If this is so, the conformer is discarded and the next most frequent conformation for the amino acid is chosen, iteratively. If the atoms of some side-chains are too close to other side-chain atoms, all the amino acids involved in the unfavorable interactions are labeled as "active residues" and clustered. This means we group the residues with unfavorable interactions among themselves, but none with amino acids outside the cluster. Their conformation can now be locally optimized by the search procedure that, for SCWRL and most side-chain-modeling procedures, is based on a dead end elimination strategy.

Briefly, the dead end elimination strategy, which is an optimization procedure that can be also used for applications other than side-chain modeling, is based on the idea that there are some values of the rotamers that are incompatible with the global energy minimum conformation. We can skip, in the search, rotamers of one residue if another rotamer for the same residue always has a lower-energy interaction energy with all other side-chain and main-chain atoms of the protein, irrespective of which rotamer is chosen for the other side-chains.

4.14
Model Optimization

The steps outlined above – template selection, construction of the main chain of the core, prediction of the loop regions, and positioning of the side-chain – provide us with an approximate model of the target protein. Ideally we would now like to optimize the resulting structure on the basis of energy calculations. In CASP experiments, it is possible to submit pairs of models for the same target, one representing the structure before optimization and the other representing the structure after optimization. The goal is to verify whether optimization procedures are able to modify the starting model to make it more similar to the experimental

structure of the target protein. So far the results have been rather unsatisfactory and no method seems to be able to consistently improve over a starting model.

4.15 Other Approaches

The step-wise procedure outlined above is, as already mentioned, not necessarily ideal for achieving a sensible model of our target protein. There are two approaches that do not strictly follow this classical strategy. The very popular method Modeller constructs the complete models on the basis of spatial constraints. In other words, it computes a set of distance and dihedral angle probability distributions that must be satisfied by the final models and then builds the models that are compatible with these distributions. The probability distributions are derived from a detailed analysis of family of proteins of known structure. For example, one can compute the probability of observing a certain $C\alpha-C\alpha$ distance in a protein, given the observed distance in a homologous protein, the type of amino acids, the dihedral angles, the sequence similarity between the two proteins, etc.

The spatial restraints and some energy terms to ensure proper stereochemistry are combined into a target function that is optimized by simulated annealing. Several different models can be calculated by varying the initial structure. The local variability among these models can be used to estimate the errors in the corresponding regions of the model.

The most recent developments in comparative modeling are based on the idea of constructing several models for each target protein and selecting the most likely only at the end of the complete model-building procedure. In other words, rather than optimizing independently each of the steps of the procedure, the most successful methods funnel into each subsequent step not only the optimum but also the suboptimum intermediate results. Selection of the final model is based on the analysis of the several resulting atomic structures. To some extent this is as if all the stages of the model building procedure (template selection, alignment quality, local templates for insertions and deletions, and side-chain positioning) were optimized at the same time rather than sequentially. The strategies for building several alternative models include selection of templates not only on the basis of sequence similarity, but also on the basis of sequence-structure fitness evaluation, taking advantage of algorithms developed for fold-recognition prediction methods. Sometimes, the template originally selected is used for searching the data base of structurally related proteins to select folds that can be used as alternative templates. Both optimum and suboptimum sequence alignments with each of the putative templates are used as the basis for model building and additional three-dimensional models are sometimes generated by combining fragments of the obtained models.

Evaluation of the final set of models, after a structure clustering step, can be based on several independent criteria, for example evaluation of local environment and inter-residue contacts, knowledge-based pairwise potentials, stereochemical quality, and, occasionally, visual inspection.

These methods can produce, on average, better models than those obtained by conventional step-wise modeling procedure, probably because it is more effective to evaluate the quality of a final three-dimensional complete model than that of each of the intermediate results of the procedure.

4.16 Effectiveness of Comparative Modeling Methods

In the last chapter we will describe some successful practical applications of comparative modeling in which a combination of methods, expertise, experimental data, and careful analysis of the model has led to significant progress in the understanding of biological systems. Here we will review what we have learned from the various CASP experiments, in which the submitted models are usually produced in a very short time and therefore reflect more faithfully the state of the art in the field.

The overall picture is that models with very respectable accuracy can be built for proteins sharing a significant similarity with proteins of known structure. A sequence identity between target and template of more than 40% essentially guarantees that the overall structure is correctly predicted by most, if not all, techniques (Figure 4.21).

When a more distant relationship exists between the protein under study and the closest protein of known structures, the gap between different methods becomes

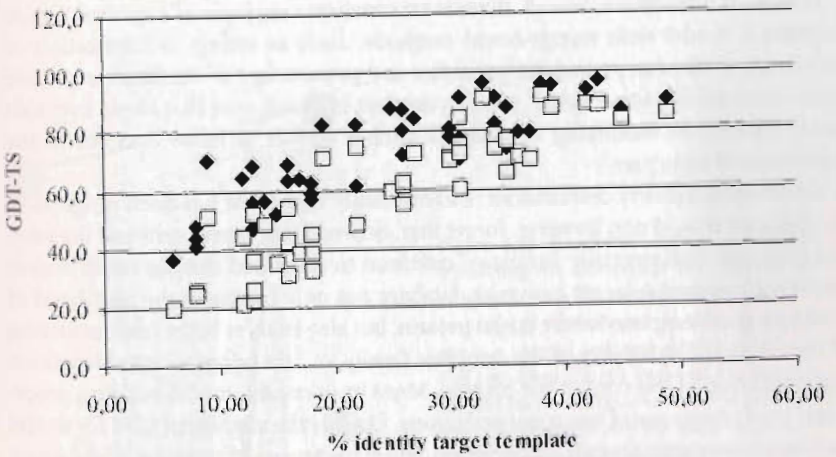


Figure 4.21 The relationship between the GDT-TS of the best (filled symbols) and average (open symbols) models and the sequence identity between the target protein sequence and the sequence of the best structural template. The data are taken from the CASP5 results and indicate that, above 40% sequence identity between target and template sequence, most methods can produce very respectable models. In more difficult examples the best methods can still produce useful results, but the gap between the quality of their results and those that can be obtained on average increases.

more apparent. As we mentioned, techniques that build several models and, only at the end of the procedure select the final method, perform better. This is not only true for "human" predictions, i.e. for predictions submitted by research groups, but also for predictions obtained by automatic servers. Some of these, called meta-predictors, use a similar strategy. Rather than building a model, these servers outsource the prediction to several other servers, collect the results and either combine or score them to provide the user with a final model. Although these methods work, on average, better than single servers, one should not forget that they can exist only insofar as "regular" servers keep being developed and improved.

Prediction of the structure of loops, especially if longer than a few residues, is still an open problem and accuracy is not yet satisfactory. What is worse, although it is possible to estimate, a priori, the accuracy of the prediction of the backbone of a protein on the basis of the sequence similarity between the target and template(s) proteins, there is no clear way to learn beforehand whether or not a given loop will be correctly predicted by any method; this is a serious limitation. The accuracy of the prediction of side-chains is rather difficult to estimate. The number of side-chains that one can include in the evaluation is quite limited. They should be those in targets solved with high accuracy by X-ray crystallography, not exposed to the solvent, with low B factors, and not involved in crystal contacts. One accepted conclusion is that the accuracy of side-chains is very dependent on the accuracy of backbone prediction. In other words, the better the prediction of the backbone, the better methods for building side-chains work. This implies that improvement of the quality of the prediction of the backbone will produce, by itself, an improvement in the prediction of the side-chains.

Finally, as already mentioned, there is no consistent example of a method able to improve a model with energy-based methods, such as energy minimization or molecular dynamics. Anecdotal examples are present in the literature and have been observed for some CASP predictions, but it is still true that these methods rarely succeed in modifying a model in such a way as to make it closer to the experimental structure.

CASP results clearly demonstrate without doubt that there has been progress in the field; we should not, however, forget that, between one experiment and the next, the sequence and structure databases continue to grow and this, by itself, makes predictions easier. A larger structure database not only increases the likelihood of finding a good template for the target protein, but also enables better understanding of the structural variability of the template family and therefore a better definition and prediction of the core of the protein. More importantly, model-building procedures rarely make use of two sequences alone. Usually the alignment used for model building is extracted from a multiple sequence alignment of proteins of the same family, and its quality will depend upon the number and the similarity distribution of all the sequences in a multiple sequence alignment. A larger sequence database enables more sequences of the target and template family to be included in the multiple sequence alignment and this is likely to improve the alignment, as we discussed. Indeed, this latter effect can be taken into account at least partially when comparing results between different editions of the CASP experiment.

For example, one can analyze the multiple sequence alignment available at the time of each experiment for each target, calculate the pair-wise sequence identity between each pair of sequences and use the values to construct a graph similar to that shown in Figure 4.22, in which each node represents one of the sequences in the multiple sequence alignment and the lengths of the edges are proportional to the distance (inversely proportional to the percentage of identity) between the connected nodes. The multiple sequence alignment is a path in the graph that includes all the sequences.

The difficulty of aligning target and template depends upon the availability of intermediate sequences, and this is determined by the most difficult pair-wise alignment that we need to perform to go from the target to the template. In other words, we might end up aligning a target and a template sequence only sharing a very low sequence identity, but we might achieve this by aligning pairs of very similar intermediate sequences, starting from the target and "jumping" from one sequence to the other until we reach the target, much in the same way as we might cross a large river jumping from one stone to the next. The difficulty of crossing the river is not proportional to its width but to the longest jump that we need to make between two stones.

Given all possible paths including target and template, we are therefore interested in those in which the maximum distance between each pairs of traversed nodes is minimal (Figure 4.22). When such a path is found, the longest edge in the path, i.e. the sequence similarity between the two most diverse sequences in the path, is an estimate of the difficulty of aligning target and template, given the distribution of sequences in the multiple sequence alignment. This approach gives,

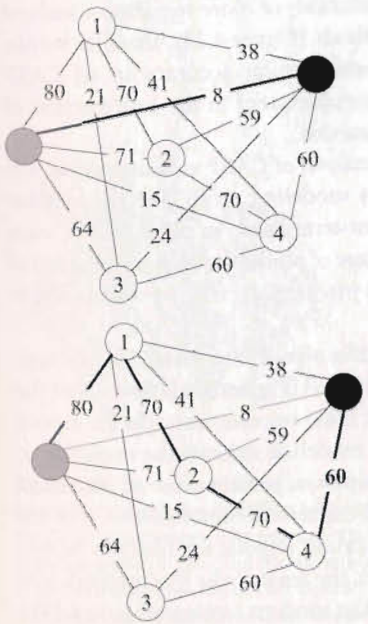


Figure 4.22 Graphical scheme of a method for evaluating the difficulty of aligning two protein sequences when a multiple sequence alignment is available. In the scheme, each circle represents a protein and each edge is labeled with the sequence identity between the two connected proteins. Assume that the gray circle is the target protein and the black circle the template. The sequence identity between the two protein sequences is only 8%. We can, however, progressively align the proteins following the path indicated by the ticker lines in the lower part of the figure. In this instance the most difficult alignment that we are forced to perform is that between the protein labeled "4" and the template sequence.

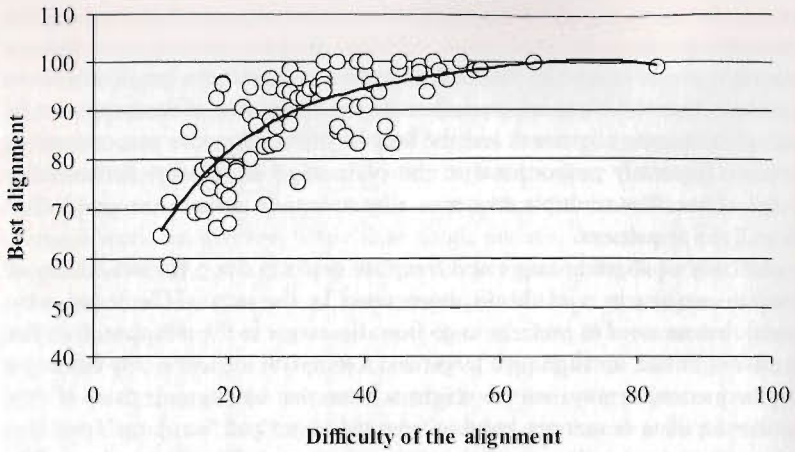


Figure 4.23 Relationship between the difficulty of aligning a target and template protein sequences, computed as described in the legend to Figure 4.22, and the best alignment obtained in the CASP experiments for the same pair of sequences.

to a first approximation, a measure of the difficulty of aligning the target and template sequence for each target in different experiments, given the database available at the time of the prediction, and can be used to ask whether the alignment of targets and templates of equivalent difficulty has become more accurate with time. The answer is that any improvement in alignment among different CASP experiments is mainly because of the availability of more sequences, and not because of a genuine improvement of the methods (Figure 4.23). In other words, targets of equivalent difficulty, are aligned with similar accuracy in all CASP experiments (in fact, in all CASP experiments subsequent to the introduction of effective methods for multiple sequence alignments).

Another frustrating aspect that arises from analysis of CASP experiments is that no method yet seems very effective at correctly modeling multi domain proteins when the domains are modeled using different templates. In other words, each domain can be correctly modeled taking advantage of its similarity with domains of proteins of known structure, but it is not easy to predict their relative orientation in the target protein.

In all the cases analyzed, the region of the active site of enzymes is, on average, better predicted than the rest of the protein structure. It is generally believed that this reflects more the intrinsic higher conservation of these regions than specific aspects of the methods. By its own nature, comparative modeling exploits the evolutionary constraints posed by the biological function upon a protein and is, therefore, expected to work better on regions that are well conserved. Nevertheless, it is still important to bear in mind that functionally important regions are likely to be well predicted by comparative modeling. This is, in fact, the reason why this method, with all its pitfalls and problems, is an invaluable tool in modern biology (Figure 4.24).

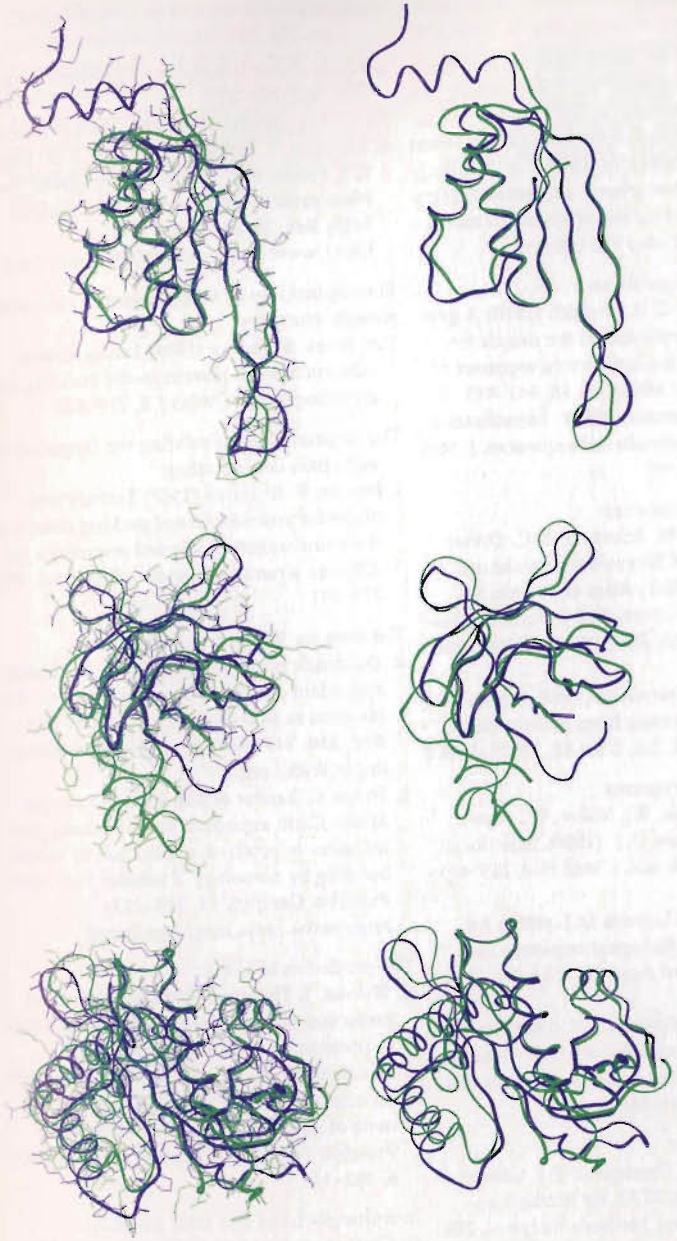


Figure 4.24 Some examples of predictions obtained by comparative modeling techniques in the CASP experiments. The experimental structures are shown in blue and the models in green in all three examples. On the left both structures are shown with their side-chains. The percentages of identity between the cores of the target protein and the best available template are 19%, 27%, and 10%, respectively. The difficulty, defined in Figure 4.22, is 26%, 27%, and 18%. Note that in all the examples the peripheral parts of the proteins are predicted less accurately.

Suggested Reading

In this chapter we mentioned several tools, most of them available via the Internet. We will give the reference to the original article. Most of the tools described here can be found at either http://ncbi.nlm.nih.gov or at http://www.ebi.ac.uk. For others, whenever applicable, the location of the server where the tool can be found is listed after the reference.

The alignment algorithms:

S.B. Needleman, C. D. Wunsch (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48, 442–453

T. Smith, M. Waterman (1981) Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197

The substitution matrices:

M.O. Dayhoff, R. M. Schwartz, B. C. Orcutt (1978). A model for evolutionary change. In: M. O. Dayhoff (Ed.) Atlas of Protein Sequence and Structure, Vol. 5, National Biomedical Research Foundation, Washington, pp. 345–358

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

Database search programs:

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990) Basic local alignment search tool. J. Mol. Biol. 215, 403–410

Pearson W. R. and Lipman D. J. (1988) Improved tools for biological sequence comparison. Proc Natl Acad Sci USA 85, 2444–2448

Altschul, S. F. and Koonin, E. V. (1988) Iterated profile searches with PSI-BLAST– a tool for discovery in protein databases. Trends Biochem. Sci. 23, 444–447

Alignment methods:

D.G. Higgins, J. D. Thompson, T. J. Gibson (1996) Using CLUSTAL for multiple sequence alignments. Methods Enzymol. 266, 383–402

C. Notredame, D. G. Higgins, J. Heringa (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. 302, 205–217

The PFAM collection of hidden Markov models:

A. Bateman, E. Birney, R. Durbin, S. R. Eddy, K. L. Howe, E. L. Sonnhammer (2000) The Pfam protein families database. Nucleic Acids Res. 28, 263–266; http://www.sanger.ac.uk/Software/Pfam/

The original idea of using fragments of known protein structures:

T.A. Jones, S. Thirup (1986) Using known substructures in protein model building and crystallography. EMBO J 5, 819–822

The original paper tabulating the frequency of side-chain conformation:

J. Ponder, F. Richards (1987) Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes. J. Mol. Biol. 193, 775–791

The tools for building side-chains:

R. Dunbrack Jr, M. Karplus (1993) Backbone-dependent rotamer library for proteins. Application to side-chain prediction. J. Mol. Biol. 230, 543–574; http://dunbrack.fccc.edu/SCWRL3.php

L. Holm, C. Sander (1992) Fast and simple Monte Carlo algorithm for side-chain optimization in proteins: application to model building by homology. Proteins: Structure, Function Genetics 14, 213–223; http://www.cmbi.kun.nl/gv/hssp/

The prediction of loops:

C. Wilmot, J. Thornton (1988) Analysis and prediction of the different types of beta-turn in proteins., J. Mol. Biol. 203, 221–232

A. Tramontano, C. Chothia, A. M. Lesk (1989) Structural determinants of the conformations of medium-sized loops in proteins. Proteins: Structure, Function and Genetics 6, 382–394

Immunoglobulins and their loops:

C. Chothia, A. M. Lesk, A. Tramontano, M. Levitt, S. J. Smith Gill, G. Air, S. Sheriff, E. A. Padlan, D. Davies, Tulip W. R. (1989) Conformations of immunoglobulin hypervariable regions. Nature 342, 877–883

A. Tramontano, C. Chothia, A. Lesk (1990) Framework residue 71 is a major determinant of the position and conformation of the second hypervariable region in the VH domains of immunoglobulins. J. Mol. Biol. 215, 175–182

Progress in comparative modeling methods is described in one paper in each of the CASP issues (http://predictioncenter.llnl.gov) and in:

D. Cozzetto, A. Tramontano (2005) The relationship between multiple sequence alignment and the quality of protein comparative models. Proteins 58, 151–157

5
Sequence-Structure Fitness Identification: Fold-recognition Methods

5.1
The Theoretical Basis of Fold-recognition

Although the number of known protein structures and sequences grows at an impressive rate, still too often we face the problem of having to infer structural properties of proteins for which no homologous protein of known structure is available. In such circumstances we cannot use comparative modeling methods and are left with a sequence, or a family of sequences, for which no structural information is available. Methods that can be used to try and infer some structural properties, for example secondary structure or pairs of amino acids putatively in contact from a multiple sequence alignment, will be discussed in chapter 7.

If, however, we analyze the dataset of available structures, we can derive another property of protein structure that can be used to produce models of unknown proteins – the folding code is degenerate in that proteins that do not seem to share an evolutionary relationship can have a similar structure.

Let us assume we compare with each other all the sequences of proteins of known structure, group them into evolutionary families and select only one protein per family. If the relationship between sequence and structure were a one-to-one relationship, each of the selected proteins would have a different architecture, but this is not observed. The relationship is many-to-one in the sense that many seemingly unrelated proteins share a similar fold (Figure 5.1).

> *Question: Are all folds equivalently used by nature?*
>
> »The distribution of folds is highly non uniform. A handful, approximately ten, are shared by a large number, about 30%, of known proteins with large diversity in sequences and functions. We call these "superfolds". Among these we find the immunoglobulin fold, shared by many domains of membrane receptors, the Rossman fold, the TIM barrel, and the hemoglobin arrangement. Although it is relatively easy to

Protein Structure Prediction. Edited by Anna Tramontano

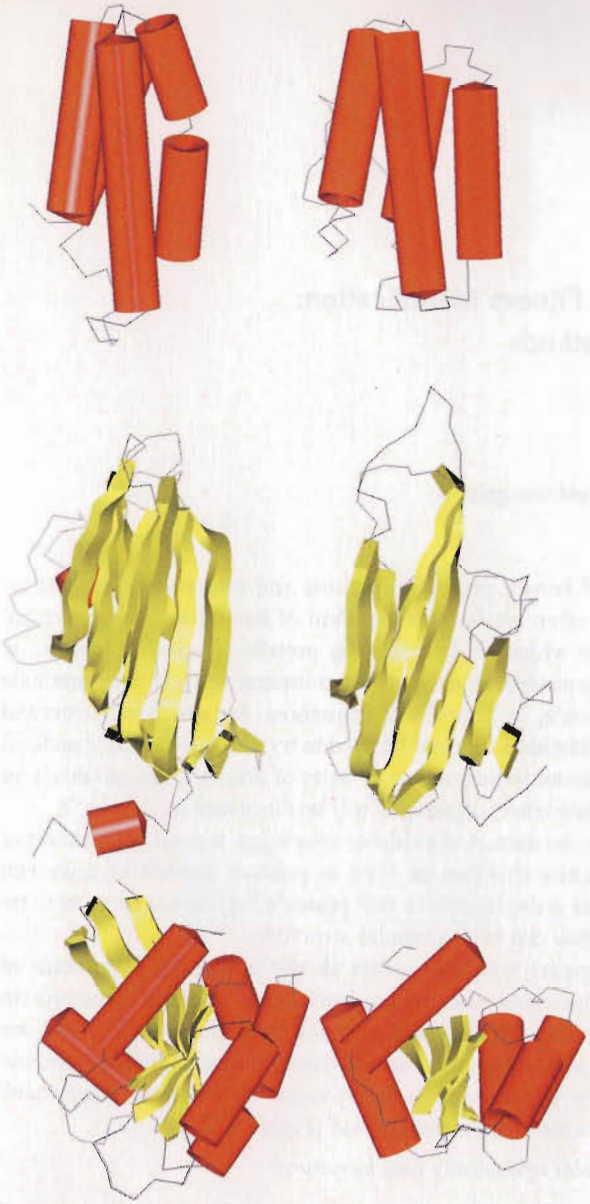


Figure 5.1 The relationship between sequence and structure is degenerate. Three pairs of apparently unrelated proteins having a similar architecture are shown in the figure. The pairs (top to bottom) are: hemerythrin (an oxygen-transporting protein) and a cytochrome B_{562} (involved in electron transport); ras p21 (an oncogene) and CheY (a protein involved in bacterial flagellum motion); a protein of the satellite tobacco necrosis virus and a tumor necrosis factor. Note that the overall topology of the proteins of each pair is similar but the size of the elements of secondary structure may differ and some peripheral extra elements can be present in one protein but not in the other.

identify close homologous relationships, as discussed in Chapter 4, it is more complex to define precisely what we mean by a "common fold", therefore establishing exactly how many "superfolds" are present is not straightforward. The fact remains that nature seems to use some structural arrangements more often than others. This might be because of the inherent thermodynamic stability of the fold and/or to the prevalence of common recurring structural motifs.«

How similar are the structures in this case is rather difficult to answer, because we have nothing with which to correlate the similarity; certainly the structures will be more different than among evolutionarily related proteins (Figure 5.1), although often one of them can still represent a useful starting point for modeling the analogous one and this is the basis of the "fold-recognition" methods we will describe in this chapter. These methods attempt to detect which, if any, of the known folds can be adopted by a target protein. Although there are many approaches to the problem, the unifying theme is that they try and find folds that are compatible with the target sequence. This is a different way of formulating the prediction problem – rather than asking what is the structure of a target protein, these methods ask whether any of the known structures can represent a reasonable model for it, irrespective of the existence or detectability of an evolutionary relationship. The problem is how to evaluate the fitness function of a sequence and a structure and, usually, this can be done using two alternative approaches: "profile based methods" and "sequence threading".

5.2
Profile-based Methods for Fold-recognition

The basic idea of profile-based methods for fold recognition is that the physicochemical properties of the amino acids of the target protein must "fit" with the environment in which they are placed in the modeled structure. For example, if a known structure is used as a template to model the target sequence, are hydrophobic residues located inside the protein? And are residues likely to be found in beta strands, for example beta-branched residues, indeed placed in a beta strand? There are complications, however. A charged residue placed in the core of the protein is clearly in an unfavorable situation, unless another residue of opposite charge is nearby and can form a salt bridge with it. In other words, we should not only evaluate the likelihood that a given residue is favored in a protein position, but also take into account the putative interactions that the residue can establish with other residues in the final structure. Although this is not taken into account in profile-based methods, they can often recognize the correct fold.

Profile-based methods code each amino acid of the target sequence according to its properties, for example secondary structure propensity, hydrophobicity, average accessibility in protein structures. Structure and accessibility propensity can be

derived from statistical analysis of known protein structures, by counting how often each residue is found in helices and strands and how often it is exposed to solvent. In other implementations, they can be obtained, for the specific protein sequence, by use of secondary structure and accessibility prediction methods.

The next step is to analyze each known protein structure, or a properly selected subset of these, and assign to each position in the structure a symbol coding its environment (secondary structure, exposure to solvent, number of hydrophobic contacts with other residues) irrespective of the amino acid that happens to occupy that position in the specific protein structure. In this way, a protein structure is encoded as a string of symbols comparable with the symbols we have used to encode our target sequence. In the sequence each symbol reflects the propensity of the corresponding amino acid for a certain environment, and therefore it is related to the specific amino acid, while in the string encoding the structure, the symbol reflects the actual environment of each position, and the same type of amino acid can be represented by different symbols in different parts of the structure (Figure 5.2).

All that is needed now is to compare the two strings in much the same way we compare protein sequences. In this procedure we must again allow insertions and deletions and assign penalty values to them, define a score and compare the observed score with the expected background distribution of scores. Methods

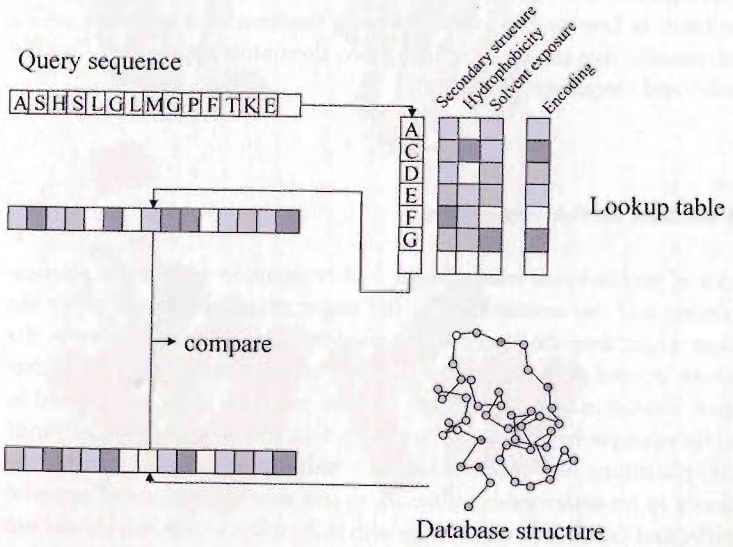


Figure 5.2 Schematic diagram of a possible profile-based method for fold recognition. The amino acids of the query sequence are replaced by a code that summarizes their hydrophobicity and their propensity for secondary structure type and solvent exposure. Each structure in the database is also encoded as a string by assigning a code to each of its amino acid positions. The code reflects their structural environment (secondary structure, solvent accessibility, and hydrophobicity of their environment). This does not depend on the actual amino acid present in the position analyzed. The string encoding the query sequence and each of the strings encoding the database structures are aligned and compared.

very similar to those used in sequence alignment are used here. The procedure can be made more sophisticated, for example by using a multiple sequence alignment of proteins of the family of the target protein and constructing a profile of propensities taking into account the variability of each position. Sometimes this procedure generates alignments between the target sequence and the "recognized" structure that point toward an evolutionary relationship, albeit distant, between the two proteins. The relationship might have been very difficult to detect when the protein sequence was compared with the whole database of known sequences, but it can become apparent when fold-recognition-based alignment points at conserved residues or regions. For this reason, in CASP it is customary to divide the targets into "homologous" and "analogous" fold-recognition targets. What the first term means is that, although it was very difficult or impossible to detect the evolutionary relationship between the target and the template on the basis of their sequences, subsequent comparison of their structures revealed common features that can be explained only by invoking an evolutionary relationship.

The distinction between comparative modeling targets and fold-recognition targets is, consequently, very fuzzy. Very sensitive methods for homology detection, for example hidden Markov models, can therefore be regarded as fold-recognition rather than alignment or database-searching techniques.

5.3 Threading Methods

The term "threading" was first coined in 1991 by David Jones, Janet Thornton, and Willie Taylor. Threading uses an atomic representation of protein structures and tries to "thread" a sequence of amino acid side chains on to a backbone structure (a fold) and to evaluate its fitness with the proposed template structure. The last step is usually based on knowledge-based pairpotentials (Chapter 3), that were indeed originally developed for this purpose, and on a solvation potential. The solvation potential is an important part of the energy function used in a threading method, because protein-solvent interactions play a major role in folding. Inclusion of solvent effects is also essential in calculations involving docking of proteins and ligand binding, and in protein recognition, engineering, and design. The precise calculation of the solvent contribution to energy is a problem, because the large number of degrees of freedom of water molecules makes the explicit simulation of water difficult. One means of estimating the solvation energy is to express it in terms of the reduction in the protein' s solvent-accessible area of folding multiplied by the solvation free energy per unit area:

$$\Delta G_{\text{solv}} = \sum_i \sigma_i \Delta A_i$$

where σ_i is the atomic solvation parameter of atom i of a given type and ΔA_i is the change in solvent-accessible surface area upon folding. The atomic solvation

parameters are atom-type parameters usually determined by least squares fitting of experimentally observed changes in free energies upon transfer of model compounds from vacuum (or a hydrophobic medium) to water. Several sets of atomic solvation parameters are available and it is still not established whether any is clearly superior to the others.

Given the energy function, we must use a search procedure that finds the optimum alignment between the query sequence and each structural template. In this process the optimum alignment is that which minimizes the energy of the new sequence in the template structure.

The pair interaction potential is non-local, so the energy corresponding to an alignment depends on all the interactions implied by the alignment (Figure 5.3). In sequence alignment the score for aligning two amino acids does not depend on the rest of the alignment, but only on the identity of the two amino acids. In threading, things are rather more complex, because we need to align the amino acids of the target to the amino acids of the template, but the score depends on the identity of the amino acids in neighboring positions which, in turn, depends on the final alignment.

Several techniques, for example double dynamic programming or Monte Carlo optimization can be used to address this issue, but the problem remains complex and computer-intensive. Some methods make use of the so-called frozen approximation in which the interaction partners are taken from the template protein rather than from the target (Figure 5.4). In other words, these methods place an

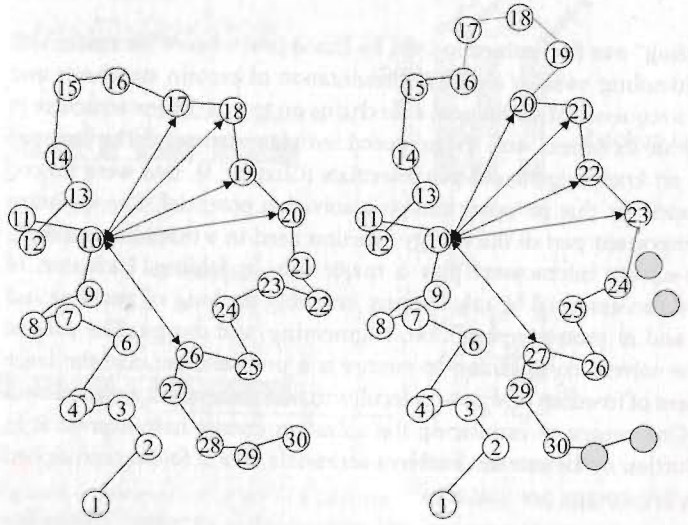


Figure 5.3 A query sequence can be positioned in a database structure in several ways, because there can be inserted and deleted residues, as shown in the right side of the figure. The interactions made by one amino acid, for example the one indicated with "10" in the figure, depend on the alignment of the rest of the sequence – the interactions of this amino acid (some of which are shown as arrows) are different in the two examples, reflecting two different alignments.

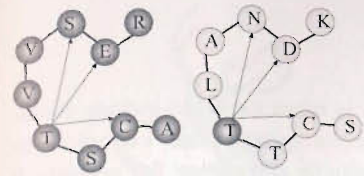


Figure 5.4 Schematic explanation of the frozen approximation. On the left, a database structure is shown with its original sequence (indicated by dark circles). In the right, the query sequence is positioned in the database structure in one of the many possible alignments. Calculation of the score should take into account which residues of the target sequence are in contact with, say, the threonine in the final alignment. In the frozen approximation, the interactions are computed by leaving the original sequence in every position of the database structure, except for the position occupied by the threonine. The procedure is repeated by substituting, in turn, each amino acid of the query sequence into a position of the target structure.

amino acid of the target sequence in a given position of the template structure and compute the energy by adding its interaction energy with the amino acids of the template. The underlying hypothesis is that, when an amino acid is placed in a given position, its interactions with the rest of the structure are approximated by its interactions with the amino acids of the template. The fitness is computed without taking into account the fact that the target amino acids will replace the template amino acids in the final model. This approximation is justified by the assumption that, if the target and template protein structures are similar, it is likely that many interactions will be conserved among them.

The procedure must be repeated for each of the templates in the selected set of folds and the resulting sequence-to-structure alignments must be sorted to decide which, if any, of the available structures is an appropriate template for the target. The results are often expressed in terms of Z-score, i.e. the number of standard deviations above the mean computed over the whole set of threading results.

If the target protein belongs to a family, it is a good idea to repeat the threading procedure using as input the sequences of other members of the family, which are expected to share the same fold. We wish to assess the compatibility of the target sequence and the template structure independently of the presence of a sequence similarity between them. It is therefore advisable to use, as input, sequences of the family as dissimilar as possible from the original target. If the same fold is found using two very distantly related sequences, our confidence in the result increases. In general, one expects to find more than one structure belonging to the same fold class in the high ranking positions. Another good indicator of correct detection of the fold seems to be a clear score separation between this first set of hits and subsequent hits.

A threading output consists of one structure, the selected template, two sequences, that of the target and that of the template, and their alignment. Careful analysis of the alignment can be used to increase the confidence in the result because, here also, there is still the possibility that a very distant evolutionary relationship between the two proteins can be detected at this stage and this makes our case for selection of the template much stronger.

5.4
Profile–Profile Methods

To improve the detection of related proteins it is often useful to include evolutionary information for both the target and template proteins, for example constructing profiles for both families and subsequently comparing the profiles. A typical scheme of a profile–profile method is to run a database search with a method such as PSI-BLAST. Usually several iterations of the program are run with an *E*-value threshold between 10^{-3} and 10^{-1}, depending on the method. Next, the significantly similar sequences are collected. Different algorithms differ in how they assign weights to each sequence and how they treat positions with a low level of sequence variation, deemed to be less informative. The simplest solution is to average all sequences from the multiple alignment with equal weights. Others perform a filtering procedure removing highly identical sequences, leaving only a set for which the similarity of all sequences is below a threshold (usually between 95 and 98%). The profile is then compared with pre-computed profiles including sequences of homologous proteins of known structure.

The methods essentially differ in the strategy used to assign scores to the comparison. Some methods use the sum-of-pairs score that we have already mentioned. A profile can be seen a set of vectors, one for each position of the alignment, and the two sets of vectors corresponding to the query profile and the precomputed target profiles can also be compared by computing their dot product or correlation coefficient.

5.5
Construction and Optimization of the Model

The final step of a fold-recognition procedure is actual construction of a three dimensional model of the target protein on the basis of the selected template and of the alignment provided by the procedure. The alignment can be optimized taking into account other factors. For example, one can predict the secondary structure of the unknown protein and compare it with the known secondary structure of the template. As we will see, the accuracy of methods for predicting secondary structure elements in proteins is quite high and the match between the secondary structure observed for the template and that predicted for the target can be used to evaluate the likelihood of the match. A word of caution is needed here, however – some threading methods use results from secondary structure prediction method to filter from the library of folds structures that are unlikely to be correct. In these circumstances, matching of secondary structure is a much less stringent criterion for the quality of the results, because it has already been used to pre-select the output.

Until a few years ago, fold recognition methods often detected the correct fold but gave a rather incorrect alignment. Because the fit of the sequence to the fold is evaluated on the basis of the alignment, this observation was rather puzzling. One possible explanation is that early fold-recognition methods were able to recognize some general properties of a fold, such as number of hydrophobic residues, sequence length, etc., rather than the specific sequence-structure combination. Matters have changed quite substantially in more recent times. By combining sequence-based methods and structure-fitness tools, current methods are rather good not only at recognizing the fold but also in producing a reasonably good sequence alignment.

When a satisfactory alignment has been obtained, the steps to follow to obtain a complete set of coordinates are the same as those employed in comparative modeling. It should be kept in mind, however, that the expectation is that the

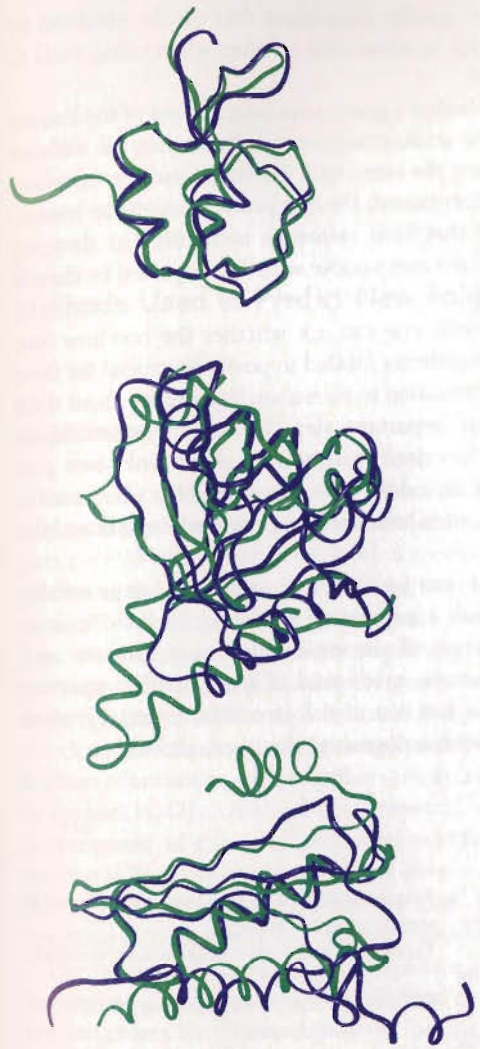


Figure 5.5 Some examples of predictions obtained by fold-recognition procedures in the CASP experiments. The experimental structures are shown in blue, the models in green. The first two proteins are examples of homologous fold recognition, the last of analogous fold recognition.

final model will, usually, be of lower quality than those that can be obtained by comparative modeling and this should be taken into account when using tools to evaluate the correctness of the model.

Fold-recognition methods detect whether a given sequence fits one of the known folds, in principle without taking into account sequence information, i.e. without assuming that the two proteins sharing the same fold, the target and the template, are evolutionary related. As already mentioned, the two proteins might be homologous, but so evolutionarily distant that their sequence similarity has dropped below the detection level, i.e. has become comparable with that expected by chance alone.

When the final model has been built, one can ask whether the residues conserved between the target and the template are located in positions crucial for their function or structure and use this information to formulate hypotheses about their evolutionary relationship. This is an important step, because detection of an evolutionary relationship might aid functional assignment of the unknown protein. Some fold-recognition methods do indeed take this possibility into account and use the sequence information to guide both the recognition of the fold and the alignment.

In general, fold-recognition methods can nowadays be regarded as quite reliable (Figure 5.5) although, in contrast with comparative modeling, it is difficult to establish a priori the expected accuracy of the model that they produce and, especially, there is no guarantee that the prediction of functionally important regions is more accurate than that of the rest of the structure, especially when the selected fold is analogous and not homologous to the query protein.

Suggested Reading

The observation that the number of folds is limited was put forward by Cyrus Chothia in:
C. Chothia (1992) Proteins. One thousand families for the molecular biologist. Nature 357, 543–544

David Eisenberg first proposed that profile based methods could be used to predict protein structures:
J.U. Bowie, R. Luthy, D. Eisenberg (1991) A method to identify protein sequences that fold into a known three-dimensional structure. Science 253, 164–170

The threading idea was described in:
D.T. Jones, W. R. Taylor, J. M. Thornton (1992) A new approach to protein fold recognition. Nature 358, 86–89

Programs for fold recognition can be found in several sites, for example:

Threader:	http://bioinf.cs.ucl.ac.uk/threader/threader.html
3DPSSM:	http://www.sbg.bio.ic.ac.uk/~3dpssm/
DOE:	http://fold.doe-mbi.ucla.edu/
123D:	http://123 d.ncifcrf.gov/
SAM:	http://www.cse.ucsc.edu/research/compbio/HMM-apps/
FFAS:	http://ffas.ljcrf.edu/ffas-cgi/cgi/ffas.pl
Robetta:	http://robetta.bakerlab.org/

The quality of fold recognition severs is automatically assessed by Livebench and Eva: http://bioinfo.pl/Meta/evaluation.html and http://cubic.bioc.columbia.edu/eva

6
Methods Used to Predict New Folds: Fragment-based Methods

6.1
Introduction

If the amino acid sequence of a protein does not reveal any evolutionary relationship with proteins of known structure and no fold recognition method proposes a putative fold with a sufficient level of confidence, we are left with the problem of predicting the structure *ex novo*. It is likely, if all the previous methods have been correctly explored, that the protein has a novel, not yet observed, fold and, therefore, we cannot use any of the topologies of proteins of known structure as a template for it. The possibility of predicting the structure of a protein in the absence of a suitable template has remained elusive for a long time. Although the chance of finding templates will increase as more structures are solved, the fraction of occasions where neither comparative modeling nor fold recognition can be applied, is still sizeable. When a new fold is discovered, it is often observed that it is composed of common structural motifs at the fragment or supersecondary structural level. This prompted the development of methods, known under the name of "fragment-based", that try and use fragments of proteins of known structure to reconstruct the complete structure of a target protein.

The relationship between local sequence and local structure in proteins is highly degenerate, and fragments with identical sequence can assume completely different structures in different proteins because of the effect of long-range tertiary interactions. A method that would simply search a database of known structures for fragments with the same sequence as those in the target protein and join these fragments would not work; we cannot, therefore, simply "fish" in the database of proteins of known structure for suitable fragments. Sequence-dependent local interactions might, however, bias the local structures that can be assumed by short segments of the chain and we can expect, for reasons similar to those described when we discussed pair potentials, that the observed distribution of conformations for a fragment can be used to derive the preference of the fragment for each conformation. In other words, the distribution of conformations sampled for a local segment of the polypeptide chain can be reasonably well approximated by the distribution of structures adopted by that sequence and by closely related sequences in known protein structures.

Protein Structure Prediction. Edited by Anna Tramontano

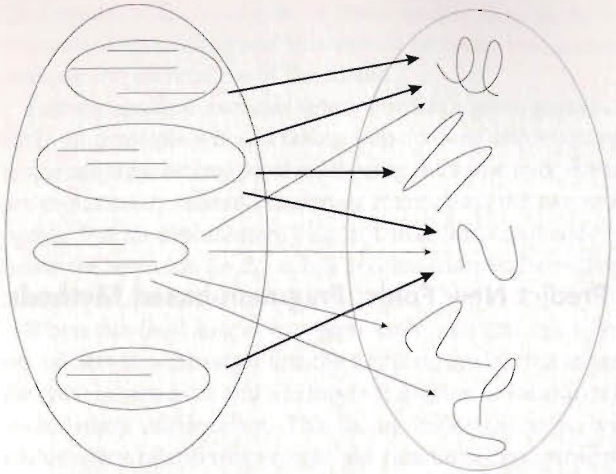


Figure 6.1 The local sequence-to-structure relationship is degenerate. For some recurring local structures, however, a correlation can be identified. In the figure, the left part depicts the space of sequence fragments. Each line represents a sequence and two similar sequences are closer to each other. Some groups of sequences will show preference for a subset of local structures (indicated by the thicker lines in the figure) while others will be less specific.

Previous studies, based on tabulations of statistics on sequences found in a given structural motif did not reveal a sufficiently significant relationship between local sequence and local structure. For example, it is well known that identical pentapeptides can be found in completely unrelated conformations. The inverse approach is more successful – by considering recurrent structural motifs and clustering them is it possible to identify sequence patterns for which a single local structure predominates. In other words, the mapping between local sequences and some recurrent local structures, such as helices, helix ends, and turns is less degenerate than for generic structural fragments (Figure 6.1).

We can use this observation to narrow the search of conformational space by preselecting structural fragments from a library of solved protein structures.

6.2 Fragment-based Methods

The overall strategy adopted by fragment-based methods is to collect the local structures assumed by short sequence segments in known three-dimensional structures and use their combinations to produce a large number of putative three-dimensional models for the target protein, among which the final model is selected on the basis of energy considerations. An important aspect of these methods is therefore the form and terms of the function used to discriminate between the many different assemblies compatible with the distribution of local

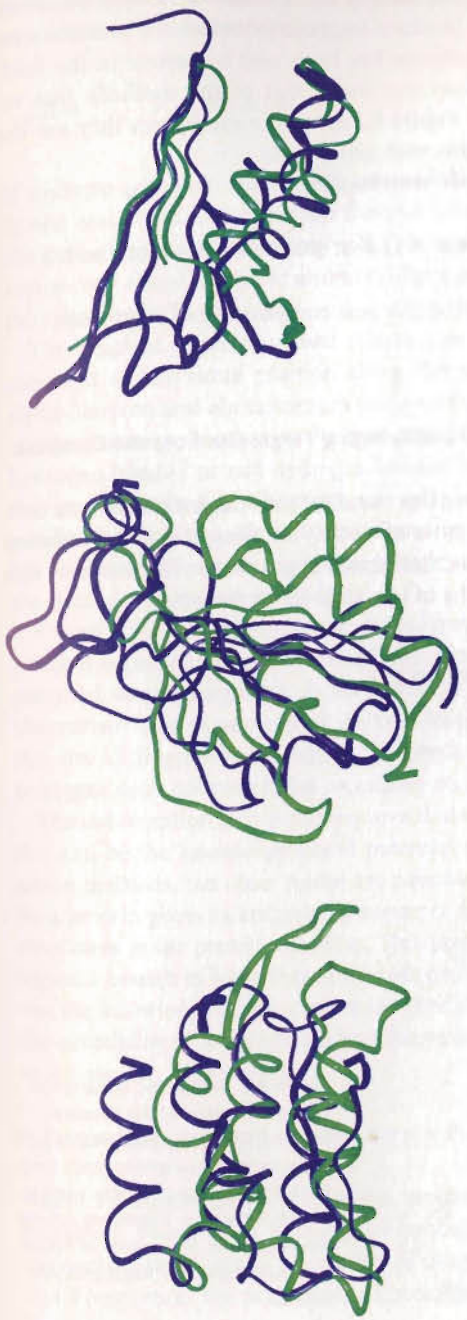


Figure 6.2 Some examples of fragment-based predictions submitted to CASP experiments.

structures. The two most popular ones are, undoubtedly, Rosetta, developed by David Baker' s group, and Fragfold, proposed by David Jones. On several occasions both methods have been shown to produce impressive models of proteins with novel folds. The success of these methods has been very important in the field, because, although their accuracy does not reach that of the methods that we described in previous chapters (see Figure 6.2 for some examples), they are the only route to the prediction of proteins with novel folds.

Given the complete sequence of our protein, they:

- split the sequence into fragments;
- for each fragment, search a database of known structure for regions with a similar sequence, called neighbors; and
- use an optimization technique to find the best combination of fragments.

6.3 Splitting the Sequence into Fragments and Selecting Fragments from the Database

There are different options for splitting the sequence into fragments. Rosetta uses nine-residue-long fragments, on the basis of a study showing that the correlation between the local sequence and the local structure is greater for fragments of this length than for other fragment lengths of less than 15 amino acids.

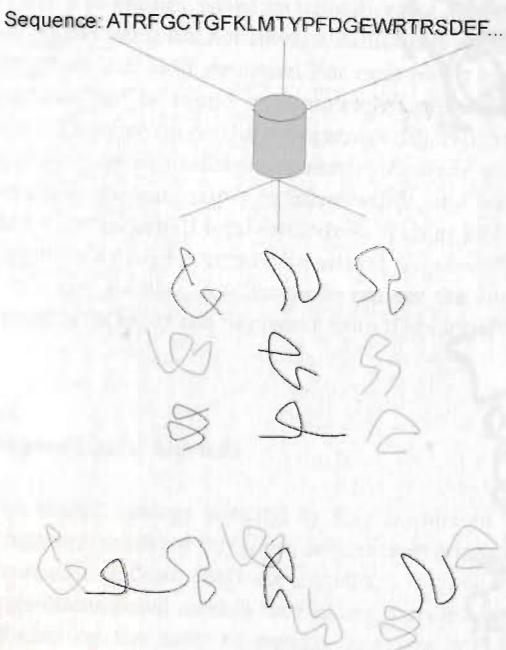


Figure 6.3 Schematic explanation of the first steps of the Rosetta method. The query sequence is split in fragments nine amino acids long. Each fragment sequence is used to search for similar fragments among the sequences of proteins of known structure. Next, the fragments are joined.

The sequence of these fragments is used to search the database of known structures for the closest 25 "neighbors" (Figure 6.3). The distance *dist* between two sequences is defined as:

$$dist = \sum_{1}^{9}\sum_{1}^{20} S(aa,i) - X(aa,i)$$

If multiple sequence alignments are available for the target and/or the template, $S(aa,i)$ and $X(aa,i)$ represent the frequencies of the amino acid *aa* in position *i* in the two alignments, respectively. If this is not true, the element of the *S* and *X* vector corresponding to the amino acid present in position *i* is set to unity and all the others to 0 (Figure 6.4).

The Fragfold method instead selects favorable supersecondary structural fragments at each residue position along the target sequence, in particular it uses alpha-hairpins and alpha corners (consecutive alpha helices in a compact or non-compact arrangement), beta hairpin and beta corners (consecutive beta strands hydrogen-bonded or not hydrogen-bonded to each other), beta–alpha–beta units that were described in Chapter 1 and split beta–alpha–beta units (two non hydrogen-bonded beta strands with an intervening alpha helix). It also uses a fragment list constructed from all tripeptide, tetrapeptide and pentapeptide fragments from the database of known structures.

For each position in the sequence, the method folds the fragment as each of the selected supersecondary structure fragments and computes the knowledge-based potential of the fragment. It then selects a predefined number of low-energy alternatives (for example ten). In one implementation, Fragfold excludes from this list all fragments for which the predicted secondary structure of the target sequence does not match the secondary structure of the fragment.

The optimization technique requires that we define the function to be optimized, this can be the knowledge-based potential described in the context of fold-recognition methods, but other routes are possible. We seek the most probable structure for a protein given its amino acid sequence and examples of sequences with known structures in the protein database. This problem can be addressed using Bayesian logics, a branch of logic that deals with probability inference, i.e. describes how to use the knowledge of prior events to predict future events. Using Bayes theorem, the probability of a structure given the amino acid sequence (and the information in the protein database) is:

$$P(\text{structure}|\text{sequence}) = P(\text{structure}) \times P(\text{sequence}|\text{structure}) \,/\, P(\text{sequence})$$

where P(structure|sequence) is the probability of observing a structure given a sequence, P(structure) is the prior probability of observing the structure, P(sequence|structure) is the probability of observing the sequence given the structure and P(sequence) the probability of observing the sequence.

```
Target                    Template
Alignment                 sequence

AGCTAVTAR                  VGCASVTAK
VGCSTFSAK
AGCTVVATK


A 200010120               A 000100010
C 003000000               C 001000000
D 000000000               D 000000000
E 000000000               E 000000000
F 000001000               F 000000000
G 030000000               G 010000000
H 000000000               H 000000000
I 000000000               I 000000000
K 000000002               K 000000001
L 000000000               L 000000000
M 000000000               M 000000000
N 000000000               N 000000000
P 000000000               P 000000000
Q 000000000               Q 000000000
R 000000001               R 000000000
S 000200100               S 000010000
T 000110110               T 000000100
V 100012000               V 100001000
Y 000000000               Y 000000000
W 000000000               W 000000000
```

$Dist_1 = |2/3 - 0| + |1/3 - 1| = 4/3$
$Dist_2 = |1 - 1| = 0$
$Dist_3 = |1 - 1| = 0$
$Dist_4 = |2/3 - 0| + |1/3 - 0| + |0 - 1| = 2$
$Dist_5 = |1/3 - 0| + |1/3 - 0| + |1/3 - 0| + |0 - 1| = 2$
$Dist_6 = |2/3 - 1| + |1/3 - 0| = 2/3$
$Dist_7 = |1/3 - 0| + |1/3 - 1| + |1/3 - 0| = 4/3$
$Dist_8 = |2/3 - 1| + |1/3 - 0| = 2/3$
$Dist_9 = |2/3 - 1| + |1/3 - 0| = 2/3$
$Dist = 4/3 + 0 + 0 + 2 + 2 + 2/3 + 4/3 + 2/3 + 2/3 = 8.67$

Figure 6.4 Calculation of the distance between the sequence of a fragment of a query protein and that of a fragment of a protein of known structure, as implemented in the Rosetta method. In the example, a multiple sequence alignment is available for the query sequence and this enables a profile to be derived for each of the nine positions. The fragment of the database in the example is instead unique and its profile only contains 1 in the row corresponding to the observed amino acid and 0 in all other cells of the matrix. For each position, the distance is computed as the absolute value of the difference between the frequency of each amino acid in the profiles of the query and database sequences. They are summed to give the distance between the two sequences.

We will use an example to explain Bayesian logics. Let us assume that we have reasons to believe (from past experience) that there is a probability of 999/1 000 that a coin is fair (i.e. that there is a 50% chance of obtaining head when we flip it) and 1/1 000 that we always get head. The probability 999/1 000 is called prior probability.

If we now flip the coin a few times, we can use the new data to re-estimate the probability that our hypothesis is true.

Let us call H1 and H2 the hypotheses that the coin is fair and that it is not, respectively. Before flipping the coin, we have P(H1) = 999/1 000 and P(H2) = 1/1 000.

The probability that the coin is fair and the outcome is a given set of flips D is:

$$P(D \,\&\, H1) = p(D)\, p(H1|D) \tag{1}$$

i.e. the probability of observing the data D (a set of flip outcomes) in the hypothesis H1 is the probability of observing the data times the probability that H1 is true, given the new data.

P(D & H1) is also equal to:

$$P(D \,\&\, H1) = p(H1)\, p(D|H1) \tag{2}$$

i.e. the probability of observing the data D (a set of flip outcomes) in the hypothesis H1 is the probability that the H1 is true times the probability to observe the data in the hypothesis that H1 is true. By combining equations (1) and (2) we obtain:

$$P(D|H1) = p(D)\, p(H1|D)/p(H1) \tag{3}$$

And, analogously:

$$P(D\,H2) = p(D)\, p(H2|D)/p(H2) \tag{4}$$

In other words, the posterior (after the observation) probability of our hypothesis is the product of the prior probability of the hypothesis times the probability of observing the new data, if the hypothesis is correct, divided by the probability of observing the new data:

Posterior = prior × (prob (new data|hypothesis H1 is true) / prob (new data)

Combining (3) and (4):

$$P(H1|D)/P(H2|D) = P(D|H1)/P(D|H2) \times P(H1)/P(H2)$$

If we now flip the coin and obtain five heads, P(D|H1), the probability of obtaining the observed data in the hypothesis that the coin is fair, is equal to $1/2^5$ and

P(D|H2), the probability of obtaining the observed data in the hypothesis that the coin is not fair, is equal to 1. Our prior probability tells us that:

$$P(H1)/P(H2) = (999/1000)/(1/1000) = 999$$

$$P(H1|D)/P(H2|D) = 1/2^5 \times 999 = 31\%.$$

We can apply Bayes logics in many fields for testing hypotheses and derive posterior probabilities of events given prior probabilities and new data.

In Rosetta-like fragment-based methods we divide our protein into fragments of a given length and search for each of the fragments in a database of known protein structures to determine the probability that the fragment is found in a certain conformation. This implies that, each time, we search for a given sequence, therefore P(sequence) is always 1:

$$P(\text{structure}|\text{sequence}) = P(\text{structure}) \times P(\text{sequence}|\text{structure})$$

Now we need to compute the prior probability P(structure) of the structure. We can assume that each structure is equally probable for our sequence and set P(structure) to 1/(Number of structures). It is, however, also useful to include extra terms since we know that low-energy folds are compact, have optimum hydrogen-bond networks, and have no steric clashes. In fold recognition, these additional terms are unnecessary, because real protein folds are almost always compact, have no steric clashes, and have well-defined hydrogen-bonding networks.

Different methods, and different implementations of the same method, use different expressions for P(structure) trying to take these factors into account. For example, one could use P(structure) = 0 for fragments with overlapping atoms and assign a P(structure) related to the radius of gyration to all other fragments, or we may take into account the orientation of local structure elements by relating P(structure) to the separation and relative orientation of local structural elements.

Question: What is the radius of gyration?

»The radius of gyration is a property characterizing the size of a particle of any shape. For a rigid particle consisting of mass elements of mass m_i, each located at a distance r_i from the center of mass, the radius of gyration, g, is defined as the square root of the mass-average of r_{i2} for all the mass elements, i.e.:

$$g = \left(\frac{\sum_i m_i r_i^2}{\sum_i m_i} \right)^2 \text{«}$$

The term p(sequence|structure) is the probability of observing a conformation given the sequence of the fragment. We can use terms related to the solvent

accessibility and secondary structure preference to evaluate the likelihood that a given sequence assumes the structure we are observing. Taking into account what we said so far, we can compute the probability that a fragment assumes a certain conformation in our final protein structure, i.e. P(structure|sequence).

6.4 Generation of Structures

To generate the set of structures for the target protein, Rosetta uses simulated annealing. It starts from an extended chain and each move consists in substituting the dihedral angles of a randomly chosen neighbor at a randomly chosen position for those of the current configuration. Conformations are initially evaluated using Bayesian derived probabilities. In subsequent cycles, knowledge based potentials are used.

Fragfold generates a random conformation for the target sequence by selecting fragments entirely randomly. Fragments are joined by superposing the α-carbon and the main-chain nitrogen and carbonyl-carbon atoms of the C-terminus of one fragment on the equivalent atoms of the N-terminus of the other fragment. If the resulting conformation has any pair of atoms closer than a predetermined minimum distance, it is rejected and another randomly generated conformation is selected by using the same procedure. This is repeated until the starting conformation has no steric clashes. When a conformation with no clashes has been obtained, the conformation is modified by randomly selecting a fragment conformation from the fragment lists. The choice is biased by the value of the knowledge-based potential of the fragments (low-energy fragments are more likely to be selected). Fragfold also uses simulated annealing. Each move is made by either selecting a supersecondary structure fragment, or a completely free choice is made from the additional list of small fragments. Half of the moves made involve a supersecondary structure fragment, and half involve a free selection from the small fragment list.

In general in these methods, several simulations are run for each target sequence and the results from one is chosen as the final model according to its energy. Sometimes the final results are clustered and the size of the cluster is used as an additional criterion for selecting the model.

The development of fragment-based methods is undoubtedly the most exciting new advance of the past few years in protein structure prediction and we are just starting to see the effect these methods are having in the biological sciences. So far, they are still rather computer intensive and this is limiting their usage. Automatic prediction servers using fragment-based methods are still few and take a long time to provide an answer, and this sometimes discourages users. They are also having an impact on structure prediction in other subject areas, for example to attempt prediction of structurally divergent regions in comparative models.

The Rosetta method has also been applied to de novo protein design. Protein design is the inverse of the folding problem. The challenge in this case is to find a

sequence able to fold into a given structure. The structure can be that of an existing protein, so that the problem is to redesign its sequence, or a completely novel fold, not yet observed. Scientists have been working on this problem for more than a decade, with variable results. Recently, in a very successful experiment, the Rosetta method was used to design an artificial sequence able to fold into a structure with a topology not yet observed in any natural protein. The impressive similarity between the designed and experimental structure, subsequently determined, shows that the goal of designing customized proteins able to fold in a predetermined way, and possibly performing a desired function, is within our reach.

Suggested Reading

The two fragment based methods, Rosetta and Fragfold, are described in:

K.T. Simons, R. Bonneau, I. Ruczinski, D. Baker (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA Proteins Suppl 3, 171–176

D.T. Jones (1997) Successful ab initio prediction of the tertiary structure of NK-lysin using multiple sequences and recognized supersecondary structural motifs. Proteins Suppl 1, 185–191

Each of the CASP issues (published every two years in the journal Proteins: Structure, Function and Bioinformatics published by Wiley) contains a section dedicated to the results of these methods.

Readers interested in protein design can also read:

B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, D. Baker (2003) Design of a novel globular protein fold with atomic-level accuracy. Science 302, 1364–1368

7
Low-dimensionality Prediction: Secondary Structure and Contact Prediction

7.1
Introduction

The local structure of a protein can be described in terms of its secondary structure, i.e. of the location of alpha helices and beta strands, as already discussed. The secondary structure of a protein can also be encoded by a linear string of characters, for example H for alpha helices, E for beta strands, and L for all other regions. Prediction of the location of secondary structure elements in a protein can therefore be described as a mapping problem in which we need to relate a string encoded by an alphabet of twenty letters (the sequence) into a string using an alphabet of three characters (Fig. 2.1), or a few more if we want to distinguish between different types of helix and different types of turn. This way of posing the problem widens the range of algorithms that can be used for prediction. For example, we can use automatic learning methods, i.e. methods that try to infer the relationships between objects by learning them from a set of known cases. In practice, the methods try to identify common features of the input values that are associated with the same output values.

The three-dimensional representation of protein structures, i.e. the list of their 3D coordinates, is not a good representation for automatic learning approaches, because similar features in different proteins have completely different "values" (coordinates), because each protein has its own coordinate system. The secondary structure, instead, is an ideal representation for mapping methods and, indeed, the simplicity of the formulation of this problem has attracted much attention since the early days of structural biology, when only a few protein structures were known experimentally.

The simplest hypothesis is that amino acids have a preference for one or other of the secondary structure elements and that these preferences can be used to predict the location in the sequence of helical and beta segments. We will shortly review the history of methods based on this hypothesis, because they are instructive examples of strategies used to extract information from experimental observation of protein structures. Before doing that, however, we should ask ourselves a basic question – suppose we could predict with infinite accuracy the secondary structure of a protein,

Protein Structure Prediction, Edited by Anna Tramontano